Sistemas de Recuperación de Información Web

TEMA INTRODUCTORIO

Índice

¿Qué es un Buscador?

Un **buscador** es un software que busca en una base de datos o repositorio documental, conforme a algunos criterios específicos.

- Tipos:
  - Directorios o índices
  - Motores de búsqueda
  - Metabuscadores

Directorios

- Sitio Web que gestiona una BD creada manualmente.
- Las URL están clasificadas en categorías.
- Características:
  - Selección y clasificación manual de recursos
  - Datos poco actualizados y poco exhaustivos
  - Resultados relevantes y páginas de calidad
  - Suelen ser temáticos
- Ejemplos

La diferencia fundamental respecto a los motores de búsqueda es que ofrecen un menor nº de resultados, a cambio están organizados por temas, es el caso de Yahoo, anteriormente lanzaba la búsqueda a AltaVista, posteriormente a Google y actualmente a Yahoo Search.
Para asegurarse de que la búsqueda se hace en la totalidad de la página, en la url ha de aparecer la etiqueta overpage.

*Decir que los directorios genéricos han caído en desuso.

Características de Yahoo!

- Catorce materias subdivididas en un número similar de subtemas. Bueno para Usabilidad.
- Se puede hacer una búsqueda general en cualquier sección o nivel. Si no encuentra resultados “salta” Yahoo!Search
- Cada resultado consiste en un título o una breve descripción.

Motores de búsqueda

- Recolección de URLs e indización automatizadas
- Muy exhaustivos
- Muy actualizados
- Manipulables
- Problemas con la calidad de los resultados y ambigüedad terminológica.

Ejemplos

Google.
Decimos que son muy actualizados porque consiste en una aplicación que automáticamente actualiza los registros, además todo aquello que es automático es manipulable, es fácilmente manipulable algo que supone una rutina.
Un ejemplo es el servidor de la universidad bib, al estar sobre Windows para los usuarios es más fácilmente manipulable y por tantos susceptible a virus.

*Distinguir además entre:
- Indización por extracción: es lo que realiza el motor de búsqueda
- Indización por asignación intelectual: lo que hace una persona.

Metabuscadores

Software que agrega los resultados de varios motores o directorios para encontrar las páginas más relevantes.
Sin base de datos propia
Optimización por tiempos de respuesta
Incertidumbres sobre métodos de combinación de buscadores, pesos, orden de resultados, …
Distintos tipos:
- Metabuscadores propiamente dichos
- Multibuscadores
- Agentes de búsqueda

Hay tantos q existen buscadores Searchenginecollosus.com

Tipos de metabuscadores

- Multibuscadores: no combinan los resultados, sólo lanzan la consulta en varios buscadores.
  - Ejemplos

Existen distintos tipo de multibuscadores en la red:
- Aquellos en los que se ofrecen los distintos buscadores en una misma web, cada uno con su cuadro de búsqueda.
  AltaVista—Buscar
  Google – Buscar
  Lycos—Buscar
- Posteriormente se incluyeron en una misma ventana, pero los resultados estaban claramente diferenciados si procedían de un motor u otro.

* Decir que no son ya muy utilizados puesto que existen motores de búsqueda que cubren ya los metabuscadores, aunque sus resultados sean menos estándar, por ejemplo: Google.

- Agentes de Búsqueda: metabuscadores instalado localmente.

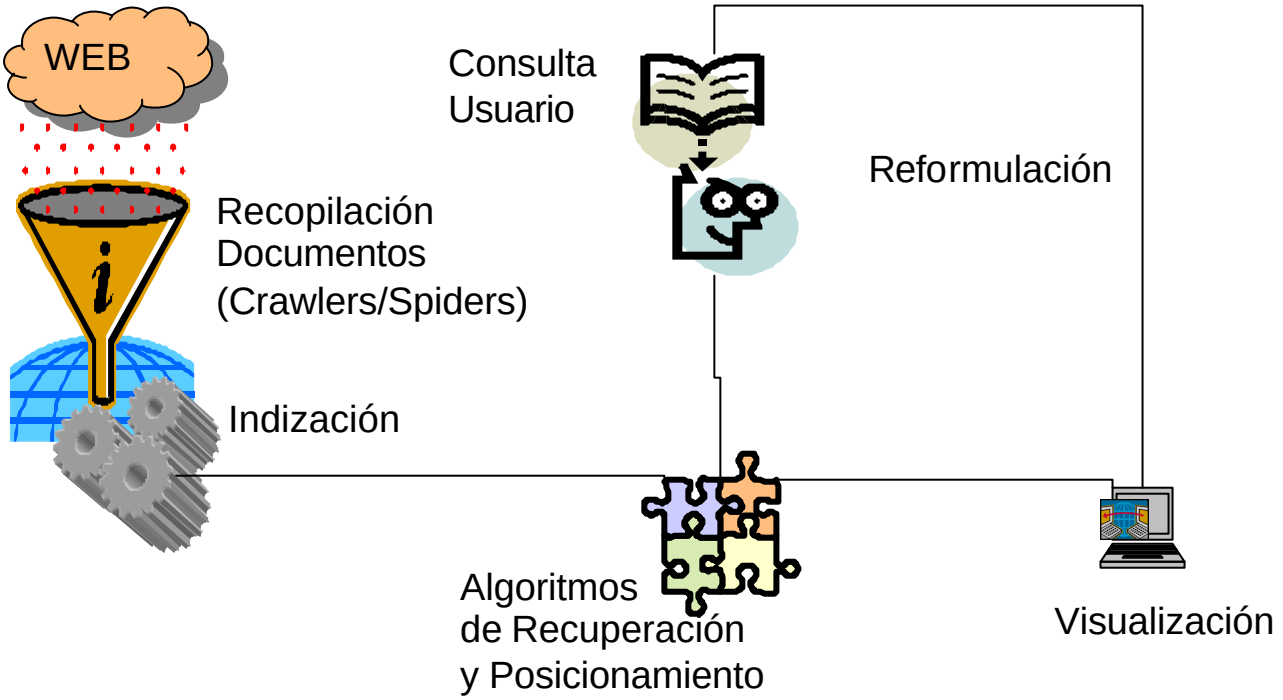
Son software, preactivos, que realizan peticiones repetidamente y periódicamente sin necesidad de enviar nosotros mismos la petición de búsqueda.

  - Ejemplos

Ejemplos Metabuscadores

Profusion http://www.profusion.com
Ejemplo es Copernic, realiza búsquedas sobre los resultados de otras búsquedas.

Motores: Sistema de Recuperación Web



Motor: Componentes

Cada Buscador tiene su propio motor: Altavista-Scooter, Lycos-Tres, Excite-Architext, Infoseek-Sidewinder, Google-Googlebot, …
Componentes:

- Spider/Robot/Crawler. Robot.txt para ver directorios permitidos. (Con robot.txt, puedo vetar los contenidos de carpetas que no quiero sean recuperadas por los buscadores, por ejemplo en una página personal, una lista de bodas que quiero sea sólo consultada por los amigos) Localizador y Recolector.

*Tienen limitaciones puesto que saltan de hiperenlace a hiperenlace, y habrá muchas páginas a la que no podremos acceder.

- Base de datos

Es una base de datos conjunta desde la que se indiza automáticamente todas las palabras del texto.

- Indizador
- Interface de búsqueda

Motores: Recopilación de documentos

- Lo realizan agentes de búsqueda (se les denomina spiders, crawlers, robots,…).
- Funcionamiento:
    1. Comienza en una página (A) y recopila todas sus URL
    2. Envía la página (A), comprueba que no está indizada y que no se tiene una versión menos actualizada, indiza la página (A)
    3. Recupera la página (B) que está primera en la lista
    4. Envía la página (B)…

Motores: Recopilación de documentos

Criterios para organizar la lista a procesar:
Puede tener en cuenta novedad o prestigio. También:

- Depth First Crawling: Hasta que no acaba con todas las páginas de un site no pasa a las del siguiente site.
- Breath Crawling: procesa primero las primeras páginas que ha encontrado en cada site, luego las segundas páginas de cada site, etc.

Bases de datos

- Actualmente existen seis grandes bases de datos:
    - Google (8000 millones de páginas)
    - Yahoo
    - MSN
    - Teoma
    - Wisenut
    - Gigablast
    - Exalead (1000 millones pero mucha información)

Los demás buscadores utilizan estas BD.

Base de Datos: Ficheros Inversos

- Permite búsquedas rápidas de textos
- Cada término asociado a un conjunto de URLs y opcionalmente a su frecuencia y posición en cada URL.
    - Suele dividirse en glosario, con la frecuencia total y número de documentos
    - Y resto de datos: URL, posición o frecuencia en el documento.
- El glosario puede estar repetido y repartido en distintas máquinas.
- Una opción para acelerar es: las páginas más populares pueden estar en muchos servidores (y a ellos se acude primero), si no hay resultados se acude a unos pocos servidores que tienen las menos populares.

Base de datos: Ficheros Inversos

Término	#d	#fr
A	99	12
Arandel	1	2
Baraja	2	2
Casa	31	40
Tecla	1	1
…	..	..
…	..	..

LEXICON

URL	Posición
uc3m.es	34, 45, 78
uc3m.es	43, 67
cartas.com	45
www.o.org	33
…	..
…	..
…	..
…	..

POSICIÓN (POSTINGS)

La base de Google

- "+"antes de una palabra no elimina aun siendo vacía, si se quiere buscar por frase poner comillas. "-" que no aparezca un término.
- No es lo mismo la ubicación geográfica desde donde hagamos la consulta (desde 2004)
- El orden de las palabras importa
- La misma consulta desde un mismo sitio con intervalo de segundos puede dar resultados distintos.
- No admite truncamiento, poner singular y plural
- No distingue mayúsculas, poner sin acentos
- Búsquedas por campos limitado
- Imposible combinar operadores booleanos de carácter distinto (todos AND todos OR pero no paréntesis)
- Aunque Google diga que existan 2000 resultados, jamás podrás pasar del resultado 1000.

Búsqueda por campos en Google

- Descubrir vínculos que le apuntan link:www.google.com
- restricciones de búsqueda de un dominio"site:ejemplodedominio.com", "site:information"
- para encontrar información de artículos de prensa en el sitio de Google: press site:www.google.com
- Para que aparezca en el título: intitle, allintitle
- Para que aparezca en la url: inurl, allinurl
- Definition:"palabra"

*Google es bueno por su algoritmo de posicionamiento más que por su algoritmo de recuperación, así cuando observamos la gran cantidad de documentos que recoge tras una búsqueda vemos que el resultado siempre es un nº redondo además, si intentamos ir a los sitios de las últimas páginas nos dice que la página no está disponible.

*Sabremos si Google ha analizado una página si en los resultados ofrece, el título y un breve resumen, de caso contrario aparecerá el enlace donde ha encontrado alguna de las palabras de la búsqueda planteada.

"Betas de siempre" de Google

- MoreGoogle (da información sobre visitas a las páginas o sobre el caché de páginas antiguas), GoogleDesktop, Barra de Google
- Google Scholar (para trabajos académicos resulta interesante puesto que ofrece análisis bibliométricos), APIS (recuerda que van contra un servidor no actual) (es una aplicación que pide un servicio a un servidor, por ejemplo Google; así para poder hacer peticiones sin que el servidor lo considere un ataque se ha de pedir un permiso, más de 1000 visitasa Google se consideran un ataque, sin un permiso Google entenderá que se está intentando manipular el servidor (virus) o plagiar el motor de búsqueda), Gview
- Calculadora
- Google Suggest, profiles y demás google labs
- Y demás Gmail (los indiza para adwords),telefonía IP,..

Yahoo!Search

Yahoo fue el primer gran directorio hecho a mano, tardo cinco años en llegar al millón de páginas, mucha calidad, poca actualización, poco exhaustivo.
2004: compra AltaVista (por contenido de su BD), AlltheWeb (por contenido de su BD), Inktomi (por su algoritmo de posicionamiento (*lo estudiaremos más adelante: cómo se ordenan los resultados) y estructura de BD), Kelkoo (como comparador de precios) y Oberture
Comienza a utilizar Yahoo Search, con el motor de Inktomi
Posicionamiento: asignación de pesos denominada WebRank (parece relacionada con la barra de búsqueda), el interfaz y otros pesos de posicionamiento copiados de Google
Yahoo es el primer buscador en número de páginas
En 2004 muchos problemas con integración de BD, se quejan mucho de la calidad resultados…mejora mucho

MSN

- Copia el algoritmo de posicionamiento de Google en 2004
- Problemas por instalación por defecto en las aplicaciones MS
- Actualmente el mejor valorado junto con Google y Yahoo

A9 AskJeeves

- A9
  - Permite buscar a texto completo en libros de muchas editoriales
  - Pertenece a Amazon
- AskJeeves
  - Ha comprado a Excite, iWon y a Teoma
  - Siempre ha tenido algo de PLN y ha establecido comparación con news
  - Actualmente el motor de Teoma hace que AJ tenga los mejores rankings de precision. Teoma utiliza un criterio denominado autoridad basado en los enlaces de las páginas del mismo tema que apuntan a la página.
  - Es actualmente el cuarto buscador mundial

Otros Raros

- Semantic Blogging Demonstrator (http://www.semanticblogging.org/blojsom-hp/semnav.html) que busca por un conjunto de metadatos más las cuestiones qué, quién, por qué, dónde.
- Exlead http://beta.exalead.com/search/
- NBII Clearinghouse http://mercury.ornl.gov/nbii// busca por metadatos.

Internet invisible: conceptos y soluciones

Internet invisible sector de sitios y de páginas Web que no pueden indizar los motores de búsqueda de uso público, por el volumen del web, la solución los metabuscadores; los enlaces; crawdler, todo lo que necesite rellenar un formulario; research Seer
Aproximadamente el 70% del Web. Con un 50% más de tráfico que el visible (mayor calidad)

- P.e., OPACs , nombres de calles en mapas de ciudades, sitios que precisen de una password,...)
- "no indizable"
  - Formato de los documentos (no son html)
  - Formularios
  - Páginas generadas de forma dinámica, imágenes, ...
  - conjunto de sitios o de páginas web que, de forma expresa, se excluyen
- No todo es Web (p.e. P2P), no todo es realmente invisible.

*El motor no encontrará todo aquello que tenga que validarse.
Google empezó a trabajar sobre esta base de datos, los recursos de Research Seer (sesgo demasiado técnico). Y aparte sacó Google Scholar Search, que se centra en documentos de tipo académicos (pdf., etc)
Google no recoge la totalidad de los documentos y los más precisos, por ejemplo todo lo recogido en las bases de datos, también lo ficheros robot.txt

Internet invisible – Deep Net

Direct Search www.freepint.com/gary/direct.htm
Turbo10 http://turbo10.com
Internet Invisible http://www.internetinvisible.com
Invisible Web http://www.invisible-web.net/
Librarian's Index to the Internet http://www.lii.org
Infomine http://infomine.ucr.edu/
Web Brain http://www.webbrain.com
Science.gov http://science.gov/
Easy searcher http://www.easysearcher.com

Criterios de evaluación de motores

Nielsen NetRankings
Actualización, tamaño, spam, enlaces muertos, cobertura según tema y área geográfica...

Actualización: Google, Hotbot, MSN y Alltheweb tardan poco, las más antiguas en Alltheweb todos pasa cada mes. MSN y hotbot los mejores. Altavista tardaba tres meses. Google parece tener el Google Dance cada 15 días y los mirrors nacionales cada mes.

Tamaño: la mayor son Google 8100 millones (101 k por pg), MSN 5000 billones (150 kpag) y Yahoo!Search 4200 millones (500k) y por último AJ 2500 millones a 101 k . Directorios manuales DMOZ y looksmart 2, 5 millones Yahoo 1,8 millones

Enlaces muertos: Enlaces que conducen a enlaces muertos, en 2000 Altavista tenía un 14% mientras que Google tenía un 4%→se hunde Altavista

Cobertura: Páginas que aparecen en un único buscador: casi la mitad están en Google (porque tiene la base de datos más grande), pero tb destacan WiseNut y Yahoo

Los más **usados** en España msn 36% google 30% terra 20%. En el mundo google 41% yahoo!Search 31 MSN 27%

Tiempo que se permanece en Google 29 m AOL 28 Netscape 13

Existe un problema de obsolescencia en la web, por lo que el motor ha de estar actualizando permanentemente los resultados, para evitar los enlaces rotos que crean insatisfacción entre los usuarios.

Criterios de posicionamiento:

- **Código de la página**

  - Metadatos

Google en principio no utilizaba etiquetas metadatos. La etiqueta META no aparece visualmente pero los buscadores acceden a estas estiquetas para la recuperación de información de las páginas.

Ej.: <META name "datescription Content (=espacio de valores)>
<META name "keyword….>
http equiv, es una etiqueta similar a la anterior, pero no es sinónima, dependerá según determinadas circunstancias el empleo de una u otra.
* Lo básico para la recuperación de información es el empleo de lenguaje controlado, es decir la utilización de etiquetas META y el uso de un espacio de valores.
Ej.: Yahoo Search parece que si utilizaba estas etiquetas para su recuperación.

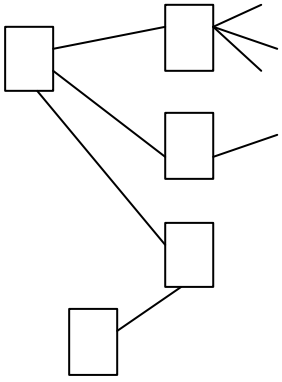
  - El formato, si tiene etiquetas de negrita, encabezados (h1, h2...) o mayúsculas, el motor de búsqueda concederá un mayor peso al sitio, pero no es acertado abusar del formato para obtener un mayor posicionamiento.
    - Accede a Tools SEO, sirve para conocer el nº de páginas que hay con determinadas consultas.
  - Descriptor externo, enlace que está apuntando a una página
  - Que el enlace sea texto, ej.: Universidad --- http://www.uc3m.es, importante porque además este tipo de enlaces son complicados de manipular.
    - Los enlaces estructurales Google los tiene en cuenta pero penalizan si sólo existen enlaces entre mis propias páginas

- **Popularidad**

  - Nº de enlaces que apuntan a una página (similar a lo que ocurre en el ámbito de la Bibliometría con el empleo de citas)
    Es lo que se denomina, Page Rank. De manera que si una página recibe 3 enlaces y una página recibe un solo enlace, el primer caso obtendrá mayor popularidad.

    A continuación un ejemplo gráfico:

¼+ ½+ ½ = 1,25



* Cuantos más enlaces reciba mejor va a posicionar el motor de búsqueda, y es que no es lo mismo que apunten a una página 100 enlaces, que uno con 4 enlaces, resulta mejor el de 4 enlaces, la razón es que existen empresas que se dedican a crear enlaces, son las denominadas Granjas de enlaces, este sistema ha sido copiado por Msn Search y Yahoo Search.

- Nº de visitas que recibe, ALEXA (aplicación que se dedica a su contabilización), se establece un baremo, por ejemplo 100.000 visitas en 3 días, por ejemplo.

➢ **Semejanza del término con la consulta**

- Frecuencia del término, si aparece por ejemplo el mismo nº de veces el término recuperación de información, recibe mejor posicionamiento aquél que recupera más veces el término en un texto menor.

- Proximidad, dependerá también de cómo de separados estén los términos en el texto, no será igual si existe una distancia de 5.000 palabras que si es de 1.000 palabras.
  Dependerá también de la significación de los términos en un contexto, así *Excite* devolvía más resultados ante una búsqueda de Banco y dinero que como Banco y peces, la razón es que existen más documentos vinculados entre Banco y Entidades bancarias que por ejemplo, Banco y parques.
  Dependerá también si las palabras clave aparecen en el título, el la URL, al principio o al final del documento, etc.

- Densidad o nº de palabras en el documento.
  Google buscaba los términos de consulta en aquellos lugares que son menos manipulables.
  Google tiene en cuenta la densidad en el título, en el URL y es que no es lo mismo que un título tenga 15 palabras a que tenga 8.

  * No se puede optimizar una página ante cualquier consulta, primero se ha de estudiar muy bien para que deseo que se recupere la página que se está creando y utilizar los criterios necesarios para que si decidimos se recupere para temas de recuperación de información, sea para esta temática y no otra.

- Clusters

➢ **Penalizaciones**

- Casos de penalización son los enlaces rotos, donde la valoración que obtiene el usuario del sitio resulta negativa (aunque no se conoce ningún estudio de este tipo en cuanto a penalizaciones); otro caso de penalización es la introducción de más de 100 enlaces en una página.
- Sand Box, es el efecto por el que un sitio sube y pasados 3 meses desaparece temporalmente del índice de Google**.** Afecta a las páginas de nueva creación y es que Google da más peso a dominios antiguos, algo que no tiene sentido para páginas de nueva creación.

- **Temporal**

  - Así una página poco actualizada contará con mayor nº de enlaces rotos. Ej.; un periódico que no actualice sus páginas es significativo de ser inadecuada para los usuarios, en cambio una página cuyo contenido trata sobre la Edad Media, pese a llevar meses sin actualizarse puede resultar excelente en sus contenidos, por lo que el criterio de actualización dependerá según sea cada caso.

Un factor que utiliza *Google* para determinar si un dominio es bueno es según el tiempo de existencia del dominio, así cuanto mayor tiempo tenga un dominio, mejor posicionamiento otorga *Google* al sitio.

- **Autoridad/Dominio**

  - Dominio

Se empezó a tener en cuanta este criterio en Google Schoolar, pero no es un parámetro normal en los buscadores.
.org – para organizaciones
.edu – en entornos académico
.com- para empresas
.es- para páginas de España

Un usuario concede una mayor fiabilidad a dominios .es y .org que a dominios .com, por ejemplo.
El precio varía según su dominio, a Google cada palabra le cuesta 2 bytes, por lo que cuida mucho la información que guarda en la base de datos, el guardar información que no se va a utilizar hace que Google funcione más lento, así que Google no almacena páginas con más de 4.000 palabras. Guarda de cada página unos 100 Kas.

*Lo que hace Google es devolver resultados de páginas sin haber analizado el sitio en su totalidad, lo que se conoce como Page rank por vecindad.

  - Profundidad en el directorio, y es que no es lo mismo
    - http://www.uc3m.es/yo
    - http://www.uc3m.es/departamento/biblioteconomia/yo

- **Criterio topográfico**

  - Siempre se ha de tener en cuenta un criterio topográfico, Google pretende darle una mayor importancia en los próximos años.
    - Por ejemplo: deseamos comprar un portátil en Internet y localizamos uno que se adecua a nuestro perfil pero se localiza en Corea.

- **Técnicas de ocultación**

Existen varias formas de ocultar el texto:

  - Que los términos aparezcan con un tamaño de letra muy pequeño, algo que no suelen penalizar los motores de búsqueda.

- Poner la letra del mismo color que el fondo, denominado ocultación o contraste fondo letra, los buscadores establecen los contrastes el problema es aun mayor cuando se utilizan hojas de estilo CCS.
- Otra técnica de ocultación es poner las palabras en etiqueta META, y es que los browser no muesran todo lo que aparece en las etiquetas META <basura>
- Google comprueba el % de código respecto al % de texto y en caso que la diferencia sea máxima piensa que se le está engañando

*Por el pago de páginas, donde entran en una especie de subasta.

Tema 1-Parte 2: Posicionamiento

Posicionamiento

- Qué es
- Factores valorados por los motores y factores indirectos
- Directrices de diseño de páginas Web
- Optimización de páginas Web para un motor
- Herramientas SEO (Search Engine Optimization)

Algoritmo de Posicionamiento

- Criterios que aplica un motor para ordenar las páginas correspondientes al resultado de una consulta.
- Son secretos comerciales
- Estrategias de Optimización, directrices de diseño aplicadas en una página para que el motor nos posicione en los primeros puestos.

Factores que influyen en el Posicionamiento

- Factores:
    - Búsqueda
    - Popularidad
    - Formato
    - Perfiles de usuario
    - Económico
    - Indirectos: Contenidos y su estructura, Credibilidad, Usabilidad y Accesibilidad

Posicionamiento de resultados

Posicionamiento por localización y perfiles
- Ubicación geográfica del que pregunta(Google topología), idioma, cookies, logs

Posicionamiento por popularidad
- Citas a tu página (Altavista, Excite, Lycos, Google, Infoseek, Inktomi) (no Nlight)
- Clic thru (hiperenlaces que se pinchan más) (todos lo tienen en cuenta)
- Popularidad relativa dentro del sitio (Lycos)
- Visitas (Alexa: page traffic: reach y pageviews)

Posicionamiento por novedad
- Mejor posición para las nuevas incorporaciones (Infoseek)
- Por revisiones (Excite), por frecuente actualización

Posicionamiento por código de página y formato
- Las Meta no mejoran el ranking en Excite, Google, Lycos, Nlight (si en infoseek y en inktomi) (en Altavista depende la fuente)
- Texto del enlace (Google)
- Formato h1..h6, negrita, tamaño grande (Google)

Posicionamiento por fiabilidad
- Si está en dominios como Gov, edu, com o net (Google)

Posicionamiento pagando
- Pay-per-click, pay-for-inclusion. (Realnames o Goto)

Posicionamiento calidad y obsolescencia

- Páginas que llevan muchos años mejor posición (Google), volumen de contenidos, calidad

Posicionamiento de resultados: Posicionamiento por similitud de Búsqueda

- Frases, presencia de frases de búsqueda en frases del documento (Infoseek)
- Palabras clave al principio y final de la página (Altavista, Inktomi)
- IDF (Altavista)
- Búsqueda booleana con OR, gana el que más palabras diferentes tiene (Altavista)
- Por clusters de conceptos asociados Excite: Intelligence Concept Extraction (tb en labgoogle.com/sets, clusty, vivisimo, exalead, kelkoo, …)
- Tamaño de página (mejor si pequeñas) (Google)
- URL con el término (Altavista, Google, Hotbot, Infoseek, Excite y Lycos)
- Posición de los términos de búsqueda al principio (Altavista, Excite, Webcrawler)
- Cercanía de los términos de búsqueda en el texto (Altavista, Lycos, Webcrawler). Google no tiene en cuenta esta densidad de palabras clave
- Frecuencia del término de búsqueda en el texto (Altavista, Lycos)
- Título, presencia de términos de búsqueda en el título (Lycos, Webcrawler)

Factores indirectos

- Usabilidad: Ejemplos
  - poner una página de bienvenida (baja la popularidad no el motor)
  - Todo lo que tenga muchos colores, no sea claro en su utilidad, se parezca a un anuncio
  - No poner las conclusiones al principio ni escribir esquemáticamente
- Accesibilidad: Ejemplos
  - Animaciones flash (es una imagen no la leen los robots), páginas dinámicas. En general cualquier requisito de software no genérico.
  - Todo lo que afecte al tiempo de descarga
- Otros: contenidos, arquitectura hipertextual y fiabilidad

Penalizaciones. Ejemplos:

Por engaños al motor y protección contra herramientas SEO

- Poner texto y fondo del mismo color (Altavista se la engañó durante mucho tiempo, Google es la única que no lo penaliza) o etiquetas o meta con contenido engañoso [Ocultación]
- Crear páginas falsas con la palabra clave miles de veces y enlaces a la página a promocionar (Google incluso no permite repetirla en la misma línea) (Infoseek) [Stuffing]
- Poner texto muy pequeño (solo Inktomi) [¿Ocultación?]
- Redireccionamientos automáticos (Altavista, Infoseek, Lycos, Excite)
- Mandar varias veces la misma página al motor p.e. en la misma semana (con algunos hay que dejar pasar tres meses). Evitar que nos lo mande un software automáticamente.
- Utilizar Cloaking (páginas falsas enmascaradas) y Doorway (páginas especiales para un buscador e ininteligibles para usuarios)
- SandBox (cuarentenas)(motivo: ¿migración BD? o ¿protección SEO para nuevos dominios?)

- Revisad http://www.google.com/webmasters/seo.html

- Por limitaciones del motor o limitaciones en su acceso
  - Tamaño (muchas penalizan el tamaño Google primeros 100k)
  - Porcentaje alto de código respecto texto, y todo lo que influye en web invisible (formularios, passwords, bases de datos, ficheros no procesables, …). Las páginas con enlaces con muchos parámetros o con sessID o sessionID
- Por limitar la comunicación con el usuario
  - Cualquier elemento que disminuya credibilidad, usabilidad o accesibilidad

Directrices de diseño

- El título
- Palabras clave
- Los metadatos
- La descripción
- El texto

Técnica de los cinco

- Titulo con 5 palabras, meta keyword las 5 palabras con comas y meta description las 5 palabras sin comas. Si está en la URL alguna de las palabras mejor aun.
- Cuerpo: empezar y acabar la página con 5 palabras clave, utiliza h1,negrita, grande...(Google), primer hiperenlace con mi palabra clave, el primer ALT de imagen con la palabra clave, usar la palabra clave en el primer tercio del texto
- Ajustar los recuentos, frecuencia...al buscador concreto (análisis de regresión, etc)
- Ajustarse al buscador: con Google intentar reducir código con respecto a texto
- En los directorios... no darse de alta automáticamente para evitar mal posicionamiento

Título de la página

- Da muy buenos resultados: Las palabras clave que definen el contenido de su página deben aparecer en dicho título y URL. Ser lo más específicos posible.
- Poner una META con TITLE (con DC.Title mejor)
- Título sencillo, acorde con el tema, con palabras clave, con pocas palabras, entorno cinco.
- Algún buscador ordena alfabéticamente, recordad que en ASCII es: " # $ % & ‘ ( ) * + , - . / 0 1 2.. 8 9 : ; <=> ? @ A …X Y Z [\ ] ^ _ ` a b c.. y z { | } ~
- Dar de alta la página pero recordad no dar de alta páginas con distintos títulos pero con la misma URL.

Palabras clave

- Pensar en el tema principal pero tb como lo buscarían los usuarios en los que nos queremos centrar
- No poner en la misma línea la misma palabra clave
- Un truco es indizar quitar palabras vacías de nuestra página y realimentar con el estudio de los enlaces que apuntan a mi página

En meta keyword

- Unas 25, valen frases (no repetir en el meta la misma más de tres veces para no ser penalizados)

- Evitar palabras muy comunes y muy raras o difíciles de escribir
- Palabras sin/con acento y en plural/singular
- Escribir mal aposta las palabras que normalmente se escriban mal (depende del idioma del público objetivo pe. Busines, bussines, ingeneering...son críticas en España)

Estrategia de posicionamiento

- Titulo con 5 palabras, meta keyword las 5 palabras con comas y meta description las 5 palabras sin comas. Si está en la URL alguna de las palabras mejor aun.
- Cuerpo: empezar y acabar la página con 5 palabras clave, utiliza h1,negrita, grande...(Google), primer hiperenlace con mi palabra clave, el primer ALT de imagen con la palabra clave, usar la palabra clave en el primer tercio del texto
- Ajustar los recuentos, frecuencia...al buscador concreto (análisis de regresión, etc)
- Ajustarse al buscador: con Google intentar reducir código con respecto a texto
- En los directorios... no darse de alta automáticamente para evitar mal posicionamiento

Descripción

- Sencilla explicación del contenido temático de su página Web
- Si se exige que tenga un número determinado de palabras, no superarlos, pueden aparecer cortadas y sin ningún sentido.
- Parece que en las meta, Google da mejores resultados si se usa el dc.description en vez de description a secas.

Consejos de creación de páginas:
Abusos con las palabras clave y Metaetiquetas

- Abusos con las palabras clave
    - No, los robots las detectan y eliminan el sitio
    - No ingrese más de tres veces una misma palabra clave
    - mantener separadas las frases y palabras que contengan palabras repetidas (Google tiene en cuenta que no aparezca en la misma línea)

- Metaetiquetas
    - seleccionar diferentes palabras clave para cada una de sus páginas
    - Tener en cuenta que las variantes morfológicas de una palabra son diferentes "casa, CASA, Casa de fin de semana"

Enlaces

Texto

- Crear sitios útiles con mucha información y escribir páginas que describan el contenido claramente y con exactitud.
- Determinar las palabras que los usuarios escribirían para encontrar sus páginas y asegúrese de que su sitio realmente las incluya.
- Sitios con una jerarquía y vínculos de texto claros.
- Agrupar todos los enlaces en un mapa del sitio, si hay más de 100 dividir la página. Google lo puede señalar como una granja de enlaces y penalizarla.
- Poner etiquetas ALT en imágenes y enlaces, el texto debe ser descriptivo si no puede ser penalizado. Aunque Google no lo tiene en cuenta el texto en enlaces pero mejora la accesibilidad.

- Comprobar que no haya vínculos rotos o código HTML incorrecto.
- El texto de los enlaces es crítico (Bombing)
- En Google poner URL absolutas. Se debe poder acceder a todas las páginas desde al menos un vínculo de texto estático.

III.4 Optimización

- Las páginas se optimizan para un motor determinado.
- Nos vamos a centrar en Google

Posicionamiento: Google

- El último gran cambio en el algoritmo es de octubre de 2004
- PageRank, cálculo basado en el número de páginas que apuntan a una página. Un enlace de una página vale más si esta a su vez recibe muchos enlaces. Tiene en cuenta 100 factores pero los enlaces son el principal dada la estabilidad de resultados a lo largo del tiempo

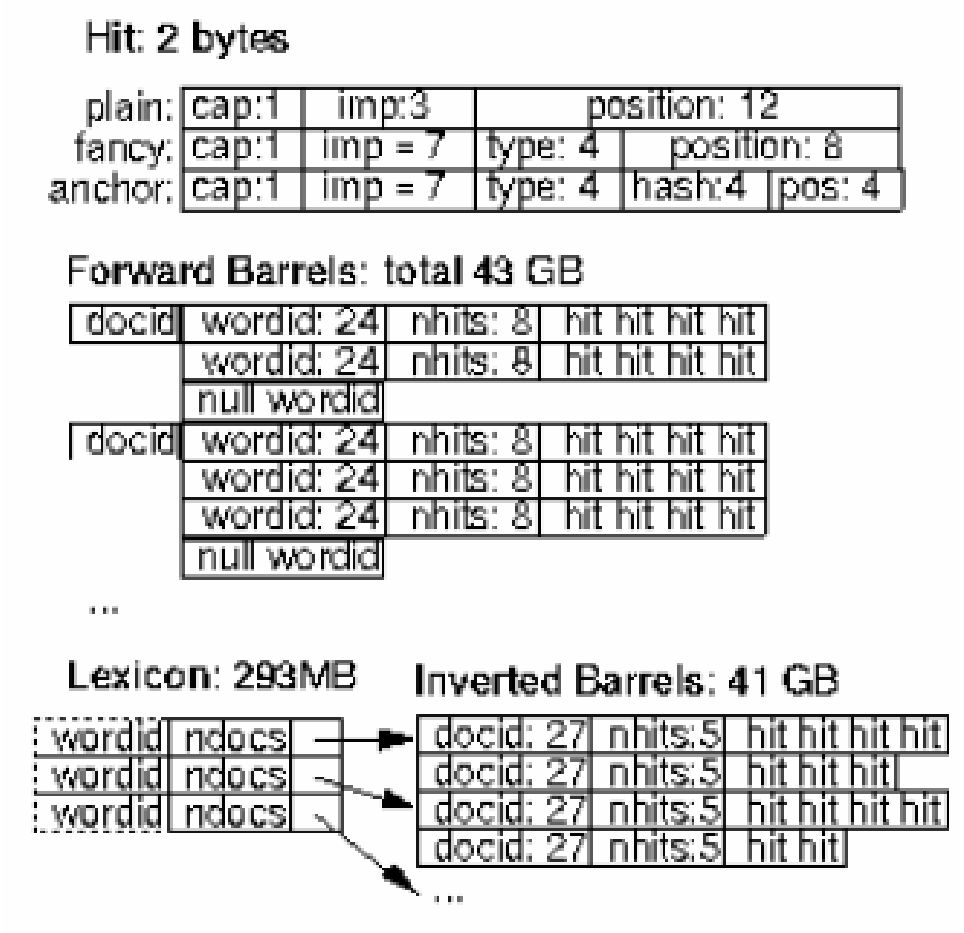
Ejemplos Google:

- Los elementos que más valora Google son:
  - Formato del texto
  - Densidad de la palabra de consulta en título y body
  - Presencia de la consulta en URL
  - Texto de los enlaces que apuntan a la página
  - Número de enlaces que recibe la página en estudio (PageRank)

Google: Base de Datos

Hits:
- Plain, los de texto
- Fancy, en titulo o URL ¿o metaetiqueta?
- Anchor: texto de enlaces

Cap: mayúsculas
Imp: tamaño letra
Position: hasta la 4096 en plain
Type: tipo de hit (titulo, URL, …)
Lexicon: fichero con el vocabulario
Inverted Barrels: documentos e
información de hits



PageRank Google

- Estimación PageRank por vecindad: Profundidad en el directorio (Google) [si el sitio principal tiene un PR de 6 y la página esta en una subcarpeta "nieta" se resta dos al PR)
- PageRank tiene en cuenta:
  - Número de páginas que enlazan a la página
  - No todas las páginas cuentan igual, los enlaces puntuan más si vienen de páginas con PR alto o páginas con pocos enlaces salientes

PageRank Google

PR(A)=(1-d)+d(PR(t1)/C(t1)+PR(t2)/C(t2)…)
PR(A), PageRank de página A
D, factor de amortiguación, probabilidad de que el usuario abandone la página. Por defecto 0,85
PR(t1)… t1, t2 son los PageRank de las páginas que apuntan a A
C(t1), enlaces que salen de la página t1, para evitar división por cero la página se considera un autoenlace.

Curiosidad:
Explicación PageRank:
http://www.webworkshop.net/pagerank.html
Calculadora de PR: http://www.webworkshop.net/pagerank_calculator.php
Saber tu PR:
http://www.googlemania.com/pagerank.php

Directrices de calidad - Principios básicos: Google

- Google, tiene en cuenta el texto del enlace con la consulta. (su manipulación se denomina Bombing)
- Tiene en cuenta enlaces internos, pero si sólo existen enlaces internos y no externos lo puede penalizar.
- Google, además, evita repeticiones en su lista de resultados.
- Crear páginas para usuarios y no para motores de búsqueda. No engañar a los usuarios presentando a los motores de búsqueda contenido distinto al que usted desea mostrar a los usuarios.
- Penalizaciones: programas de comprobación de rankings, programas para remitir páginas, intercambio de links…

Si….

- Tienes una página con frames dependes de los links externos para posicionar, usa noframe y pon buen título
- Tienes una página flash, google no la leera, yahoo si. Utiliza <noembed> para poner texto y un buen título
- Javascript, más de lo mismo, usa <noscript>

Yahoo!Search

- Incluirse en DMOZ y directorio YAHOO
- El texto de los enlaces que van en una página no son tan relevantes para esa página como lo es en Google. Poner pocas palabras en el texto de los enlaces. Enlaces y URLs dinámicas funcionan mal
- No tiene en cuenta palabras vacías (google si).
- Yahoo! Valora el texto no los enlaces que apuntan y la estructura del web (tema del site etc)
- Todas las prácticas indeseables se las traga, spam con keywords abusivos (spam interno) el spam externo funciona bien pues le da igual de quien recibe los enlaces
- Mucho peso al título, permitiendo repeticiones de palabras, se puede llegar a 100 caracteres.
- Las keywords en la URL parecen tener más peso.
- Utiliza etiquetas meta pero las da poco peso
- Tiene en cuenta enlaces entrantes
- Tarda mucho en subir una página y penaliza sin sentido

III.5 Herramientas SEO

- Los estudios de optimización son complicados, lo mejor es simplificarlo con herramientas SEO

- Herramientas de análisis:
    - Comparar Palabras clave y título con los mejores
    - Estudio de enlaces
    - Ver con los ojos de un motor
    - Monitorizar nuestra página Web y su competencia

- Herramientas de mejora:
    - Elección de título, elección de palabras clave
    - Políticas de Intercambio de enlaces
    - Monitorizar nuestra página Web y su competencia

Monitorizar páginas Web

- ALEXA http://www.alexa.com/

Ranking basado en popularidad, mide el tráfico (con usuarios con la barra de Alexa) media geométrica de:
Reach (alcance): media del número de usuarios (visitas) que recibe un SITE (no URL) por millón y día (con medias mensuales y 3m)
Page Views: número de páginas vistas por diferentes URL que han solicitado determinada página. [Diferentes=mismo día]

Limites: Alexa sólo funciona con Explorer y Windows, sólo tiene en cuenta si hay más de mil visitas (más de 100.000 en ranking), no accede a https, solo mide la URI principal no las páginas secundarias del site.

- Recuerda poner un contador o usar un analizador de logs
- Monitoriza tu página en Google con Google Monitor http://www.googlemania.com/monitor.php

Simuladores de motores

- http://www.1-hit.com/all-in-one/tool.search-engine-viewer.htm
- http://www.delorie.com/web/ses.cgi
- Search Engine Spider Simulator. http://www.webconfs.com/search-engine-spider-simulator.php
- http://www.searchengineworld.com/cgi-bin/sim_spider.cgi
- http://www.webconfs.com/search-engine-spider-simulator.php

Palabras clave

- Keyword Counter - Keyword Frequency Analyzer. http://www.keywordcount.com/ para calcular la densidad de una sentencia de búsqueda
- http://www.forobuscadores.com/recursos/index.php?m=c&c=27

Enlaces

- No siempre coincide con el número real, puede tener que ver con el cálculo del PR (las páginas apuntadas que no apuntan son eliminadas en algunos cálculos)
- http://www.webmaster-toolkit.com/search-engine-optimisation-tools.shtml
- IBP
- Link popularity check
- En Google mira los links con Google Tracking http://www.googlemania.com/tracking.php

Soluciones integradas

- IBP

Los otros medios

- Intercambio de enlaces - PageRank 6 http://www.seohome.com/intercambio-de-enlaces.html
- Rastreador.com - Intercambio de enlaces http://www.rastreador.com/intercambiodeenlaces.html
- Free for all http://www.global.gr/mtools/linkstation/ te publican tu enlace a cambio de enviarte correo.
- Aunque la regla de los cinco da buenos resultados Google podría en breve penalizada, escribir en guestbooks y weblogs no está penalizado todavía pero puede estarlo en breve.
- Pueden penalizarte si recibes enlaces de páginas poco éticas, pero nadie dice nada si son págs con alto PR

Los factores indirectos

Todos estos factores tienen un equivalente en creación de aplicaciones

- Accesibilidad
- Usabilidad
- Credibilidad
- Contenido
- Interoperatividad

Accesibilidad

Significa no poner barreras a tu público objetivo (ciegos, daltonicos, torpes, mancos, desmemoriados, propietarios de hardware obsoleto (hermanos pequeños, tercer mundo, …), niños, conexiones lentas…). Distinto de Usabilidad, es decir que puedan hacer las cosas no es lo mismo que usabilidad que es que tenga facilidad de uso.

Accesibilidad

- Se mide con test (http://www.w3.org/TR/WCAG10/full-checklist.html)
- O con programas: Bobby, TAW (http://www.tawdis.net/), http://www.cynthiasays.com/
- Existen varios niveles (W3C) A, AA, AAA

Corrección HTML y CSS:
http://validator.w3.org/ Valid XHTML 1.0
http://jigsaw.w3.org/css-validator/ Valid CSS

Usabilidad

Significa facilidad de uso y de aprendizaje
Mejora si eliminamos lo superfluo (lo cual no es lo mismo que poco estético), cumpliendo estándares tecnológicos no se es más usable.
La Usabilidad evalúa 5 propiedades:

- Facilidad de aprendizaje
- Facilidad de recuerdo
- Eficiencia usuario
- Tasa de errores

- Satisfacción

(son interdependientes, bajar una puede aumentar otra)

Calidad de uso

- Extendida a software como calidad de uso (ISO 9241-11(1998), 14598, ..)
- Es el planeamiento y diseño de un test de usabilidad con el propósito de garantizar la calidad del software.
- Se mide con test de usabilidad:
  - Exploratorios: con usuarios y con especificaciones de usabilidad.
  - Diseño con comparaciones usabilidad con versiones y productos (prototipado)
  - Evaluación final: heurística expertos (3) y tests usabilidad

Credibilidad

Disciplina que estudia como se relaciona credibilidad y: nivel de compromiso, enlaces recibidos y ofrecidos y diseño en web y software.
Se pensaba que dependia de: mención de autoría, institución, política de confidencialidad, fecha actualización,…
Estudios de usuarios revelan que está en función de la usabilidad y de calidad ténica del sitio (que funcione rápido y que impresione)

Credibilidad Positiva

De mayor a menor:
(factores por importancia >1.5 entre -2/+2)

- El sitio te ha servido en el pasado
- Prestigio de la organización que avala el sitio
- El sitio responde rápido al cliente
- Se facilita la dirección física y el teléfono
- Se ha actualizado recientemente
- El sitio parece diseñado por profesionales
- La usabilidad es correcta (contenido, comprensión, arquitectura)

De mayor a menor:
(factores por importancia >1.0 entre 1.5 (rango -2:2)

- 1.5 Dar la dirección de correo
- 1.5 organización racional (Usabilidad)
- 1.3 listar autores, existencia de citas y refencias externas
- 1.3 sitio enlazado creible
- 1.2 se muestra la política de privacidad
- 1.2 el sitio manda correos de confirmación tras las trasacciones
- 1.2 existe un buscador interno
- 1.0 recomendación de un amigo
- 1.0 permite imprimir fácilmente
- 1.0 dar la información en más de un idioma

Credibilidad Negativa

De más negativo a menos negativo:
(factores por importancia <-1.5 entre -2/+2)

- Dificultad para distinguir anuncios de texto
- Poca actualización
- Tiene pop-ups con anuncios
- Poca usabilidad (dificultad de navegar)
- Los enlaces que salen de la página son malos
- Algunos enlaces o recursos no son accesibles

De más negativo a menos negativo:
(factores por importancia entre -1.5 y -1.0 (rango -2 a 2)

- -1.4 dificultad para navegar
- -1.4 enlaces no creíbles o erróneos
- -1.3 errores tipográficos
- -1.3 el sitio no está siempre accesible
- -1.1 no coincide nombre de empresa y nombre de URL
- -1.0 se tarda mucho tiempo en descargar la página

Contenido

- Arquitectura de la información
- Diseño de páginas
- Redacción
- Legibilidad
- Usabilidad Multimedia
- Arquitectura de enlaces

CONTENIDO. Texto y Usuarios

- Los datos son para siempre el diseño es temporal. Los usuarios acceden por el contenido no por el diseño.
- Un primer vistazo debe hacer intuir el contenido.
- Análisis de audiencia:
    - Leer en pantalla es un 25% más lento q en papel. Ralentiza la lectura el uso de mayúsculas y cursiva (se parecen más las letras).
    - El 80% lee a saltos.
    - Por párrafo se lee la primera línea
    - El 50% se salta las partes de la página sin cargar.
    - **Los usuarios ignoran todo lo que se parezca a un banner.**

DISEÑO DE PÁGINA

Espacio <20% Publicidad (-posible)+ Controles Navegación
Espacio >50% Contenido

- Usar blancos para agrupar la información y focalizar (mejor para leer que las líneas)
- Páginas y tablas autodimensionables.
- La parte alta de la página sin imágenes, con texto.
- Diseño pensando en las distintas variables: letras instaladas, tamaño monitores,..:
- Distintos elementos en distintas pantallas: con distintas css, imágenes con distintas resoluciones, sin textos en gráficos (problemas traducciones, transmisión, etc.)
- Poner versiones para imprimir (600px son 8,3 pulg y una hoja 8 pulg.)

- Evitar HTML no estándar para que se interprete igual en todos los browsers (validarlo)
- Tiempo de carga de una página: si es >10 segundos los usuarios abandonan, deseable un segundo (con RTB 10 seg. son aprox. 50 kb).

REDACTAR UN TEXTO

- Más breve (50% menos texto que en cualquier otro medio). Evitar palabras no específicas del contenido como "bienvenido", texto subjetivo, énfasis, metáforas, sin juegos de palabras (usuarios multilingües), sin palabras ingeniosas ni símiles. No poner URL, sino texto, para navegar a otra página mediante clic.
- Muy estructurado: párrafos cortos, una idea un párrafo, con varios niveles de encabezados, viñetas si anidadas mejor, páginas cortas y vinculadas, subtitulos informativos más que atractivos, tipografía/colores en palabras claves (sin exagerar), ..
- Al principio del documento poner las conclusiones. Lo más importante delante.
- Lo que interesa a menos usuarios (p.e. detalles técnicos y específicos) en páginas independientes
- Se citan menos páginas con muchos temas distintos, hacer contenidos específicos.
- No dividir un mismo texto en páginas con "continuación"

LEGIBILIDAD

- Selección de colores: contraste letra fondo alto (negro con fondo blanco, lo contrario ya ralentiza la lectura). Evitar imágenes de fondo, mejor los fondos claros.
- No usar marquesinas ni realizar cambios de tamaño. El cerebro lo asocia a publicidad (banners).
- Alinear a la izquierda.
- Sans-serif (Verdana) es la más legible en tamaño 10
- No usar tablas anidadas.

HOJA DE ESTILO

- El formato siempre en hojas de estilo.
- Dan coherencia, mejoran aprendizaje y memoria del sitio
- No usar más de dos fuentes y poner alternativas
- La página debe de ser legible si desaparece la hoja de estilo.
- Tamaño de letra en tamaño relativo (%) a la que tiene por defecto el usuario.

PROGRAMAS Y MULTIMEDIA

- Videos: poner una foto estática. Poner buen sonido, los usuarios piensan que mejora la imagen.
- Imágenes: no superfluas, con algo de contexto (no recortar al máximo). Una imagen no sustituye una descripción ni una comparativa. Evitarlas como fondo y si se usan reutilizarlas en el site (cache).
- Descargas informadas (tamaño, tiempo, etc). Todo lo que supere los 10s se debe avisar.
- Siempre es mejor usar tecnología de hace dos años (mejora la accesibilidad y está más depurado). Ahora las versiones cambian poco y se adaptan lentamente (en 2 años, 90%).
- Animaciones: pocas, sólo esenciales. Se asemejan a banners (ignoradas). Se usan con éxito en escalas de tiempo, transiciones, 3D, ampliar la pantalla.
- Probar en distintos browsers y en versiones de hasta hace dos años (pe reproducir errores).
- Usar texto ALT siempre que sea posible

PAGINA DE INICIO

- Hacer visible logo y nombre de la compañía (superior izda)
- Accesible desde el resto de las páginas.
- La gente busca algo concreto y se necesita navegación rápida y fácil. Poner directorio temático, interfaz de búsqueda y enlaces a noticias especiales.
- Ancho variable. Si se exige tamaño fijo poner 600x640

NAVEGACIÓN

- Se debe sabe dónde estoy, dónde he estado y dónde ir en relación al Web y al sitio.
- Evitar navegación lineal (en recursos educativos si funciona)
- Interfaz de navegación lo más estándar posible
- Evitar nuevas ventanas, impide ir hacia atrás y agobia al usuario.

ENLACES

- Seleccionar mucho los enlaces See-Also.
- No cambiar los colores típicos. Colores distintos para visitados y enlaces.
- Poner texto en enlaces (diferente a la URL)
- Siempre deben de tener un enlace a la página de inicio (enlaces estructurales).

FRAMES

- Evitarlos (problemas para imprimir, para algún buscador, etc).
- Si se usan poner siempre el NOFRAMES . No ponerlos bordes ni scroll.
- Se pueden usar: en diccionarios largos, metapáginas (una página que muestra otra de ejemplo)

Interoperatibilidad

Metadatos
Ontologías
    RSS
    RDF
Mapeados (Alineaciones)

Tema 2: Evaluación mediante Medidas de Recuperación

Evaluación de un Sistema de Recuperación

CONTENIDO

- Cobertura
- ***Tamaño***
- Novedad
- Actualización

RECUPERACIÓN

- **Algoritmo Recuperación**
- **Algoritmo Posicionamiento**

Recall
Precisión

DISEÑO

- Interfaz de búsqueda
- Arquitectura:

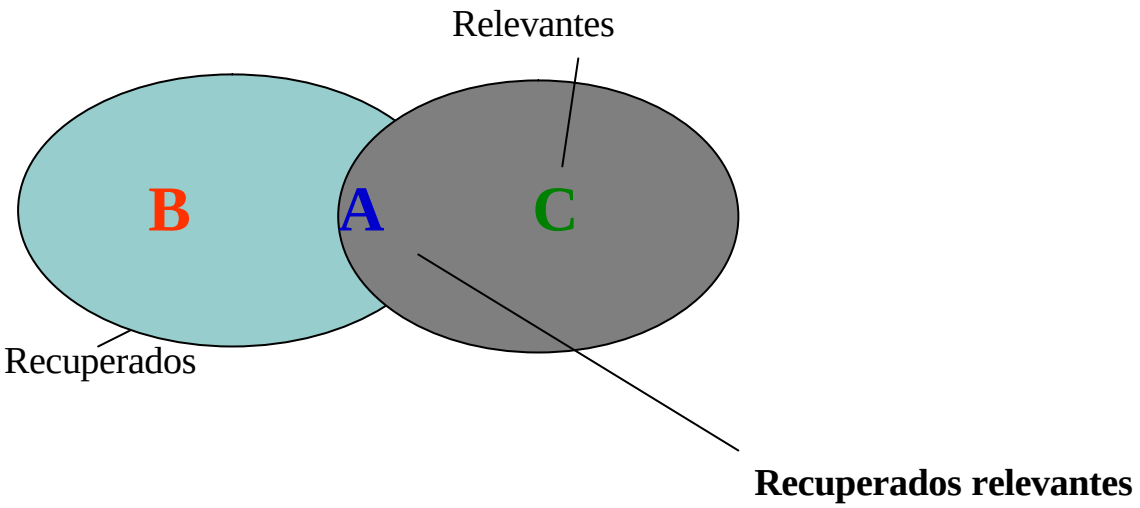
{ Estrc.índices (árboles, hash, ...)
Tipo almacenamiento datos, etc
Eficacia almacenamiento
(Indices+reg.doc)/espac.docs
- ***Eficacia de ejecución***
Tiempo en hacer una operación

- Visualización resultados
- Política de Indización

Ruido y Silencio

	Relevante	**No Relevante**
Recuperado	**A**	**B**
No Recuperado	**C**	***D***

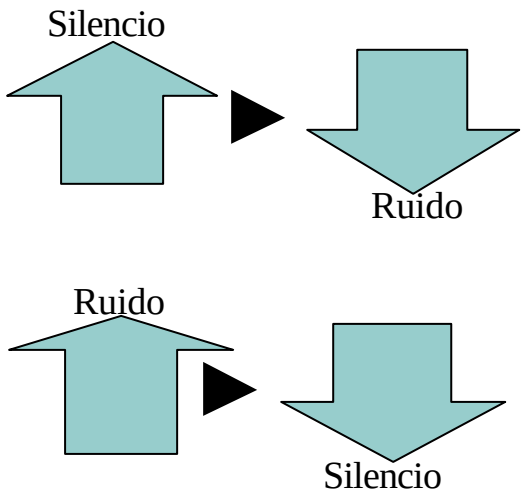
- Ruido: Documentos no relevantes recuperados (B)
- Silencio: Documentos relevantes no recuperados (C)



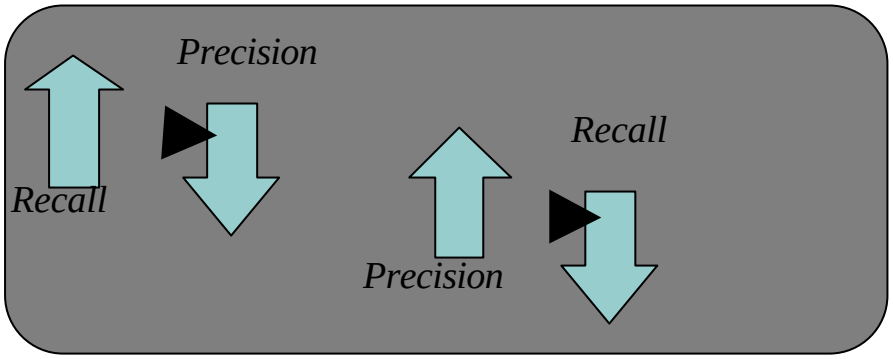
Relación Ruido/Silencio y Estrategias de búsqueda

- Disminuir Ruido
  - Consulta
    - Utilizar términos específicos, añadir términos asociados
    - Operadores AND y NOT
    - Búsqueda por frases, campos, paréntesis, evitar términos polisémicos, usar términos poco frecuentes
  - Medio
    - Utilizar Directorios
- Disminuir Silencio
  - Consulta:
    - Emplear OR, variantes ortográficas (incluido acentos, mayúsculas, género, número, ..), idiomáticas y dialectales
    - Expansión de búsqueda: Términos genéricos y sinónimos
  - Medio
    - Metabuscadores y Motores

Relación Ruido/Silencio



	Relevan.	No Relev.
Rec.	**A**	**B**
No Rec.	**C**	***D***



Recall =Exhaustividad= A/(A+C)
Mide como evita el sistema el silencio
Entre 0 y 1, mejor si próximo a 1

Precision= A/(A+B)
Mide como evita el ruido
Entre 0 y 1, mejor si próximo a 1

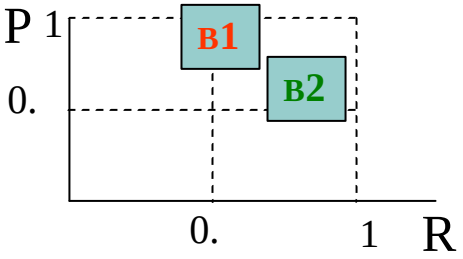
Ejercicio 1

Dos buscadores con misma consulta y misma BD
Buscador 1 r, r, r, r, r, r
Buscador 2 r, nr, r, r, nr, r, r,nr, r, nr, r, r

Donde nr es un documento no relevante y r es relevante
La base de datos tiene 10.000 documentos, 10 son relevantes a la consulta estudiada
Indicad que buscador evita mejor el ruido y el silencio según las tasas de Precision Recall

Pb1=6/6=1
Pb2=8/12=0.6

Rb1=6/10=0.6
Rb2=8/10=0.8



Ejercicio 2

Suponga los siguientes resultados de dos buscadores en Internet ante la misma consulta y la misma base de datos
Buscador 1 1, 2, 3, nr, 18, 12, nr, 4, 5, nr
Buscador 2 1, 3, 2, 18, 9, 29, 6,nr, nr, nr
Donde

- nr es un documento no relevante
- Los números son el orden de relevancia del documento
- El orden es en el que han ido apareciendo los documentos

Calcular las tasas de Precision/Recall

Solución Ejercicio 2

	Precision	**Recall**
Buscador1	7/10	7/x
Buscador2	7/10	7/x

¿Son entonces iguales los dos buscadores?

No.

Precision Recall- Problemas

- Una sola medida de precision recall mide la calidad del algoritmo de recuperación no del algoritmo de posicionamiento

(el posicionamiento solo tiene sentido cuando el modelo de recuperación lo permite)

- En Internet es imposible saber cuantos documentos relevantes existen a una pregunta dada
- No se tiene en cuenta el ajuste a la medida manual de la relevancia
- No se tiene en cuenta la interacción con el usuario
- Son dos medidas de una misma cuestión, hay que decidir a cual se la quiere dar preferencia

Precision-Recall unificada

- Medida de la F
    - Unifica Precision-recall en una única medida utilizando la media armónica, cuanto más próximo a uno mejor (a cero peor). Se mide en el j documento recuperado.

$F(j)=2/((1/r(j)+1/P(j))$

- Medida de Evaluación
    - Como la armónica pero configurable, si b>1 más peso a la precision, si b<1 a la recall

F(j)=1+b2/((b2 /r(j)+1/P(j))

Otras medidas:

- Índice de irrelevancia

Nº documentos no relevantes recuperados / nº documentos no relevantes en la colección

    - Da información aun cuando no hay documentos relevantes (¡para Recall division por cero!) o cuando no recupera documentos relevantes. Tiene en cuenta D el número de documentos irrelevantes recuperados. Cuanto más pequeña mejor
- Recall de documentos relevantes únicos (URR)
    - Sirve para comparar dos buscadores se tienen en cuenta sólo los relevantes no duplicados en los resultados de los dos buscadores
    - Nº de relevantes únicos/número total de relevantes

Gráficos de Precison Recall

- Es el sistema más utilizado en la literatura para mostrar el funcionamiento de un motor o varios
- Sirve para mostrar gráficamente, de forma sencilla, la eficacia y eficiencia de un sistema de recuperación
- Se mide la Precision a 11 niveles de Recall:

0%, 10%, 20%, ...70%, 80%, 90%, 100%

- Si no se posee determinado valor de Precision se interpola con la Precision correspondiente al siguiente Recall conocido (incluido el caso del 0% de Recall)
- Opcionalmente se puede ver la Precision en valores fijos. P.e. Cuando se han recuperado 10, 20, 30... documentos relevantes

Gráfico Precision Recall

Relevantes	10															
Recuperados	15															
		Buscador														
		r	nr	r	nr	nr	r	nr	nr	nr	r	nr	nr	nr	nr	r
Documentos Rec		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Relev Rec		1	1	2	2	2	3	3	3	3	4	4	4	4	4	5
				2/10					2/3							
Recall		10%	10%	20%	20%	20%	30%	30%	30%	30%	40%	40%	40%	40%	40%	50%
Precision		100%	50%	67%	50%	40%	50%	43%	38%	33%	40%	36%	33%	31%	29%	33%

Gráfico					
	10%	20%	30%	40%	50%
	100%	67%	50%	40%	33%

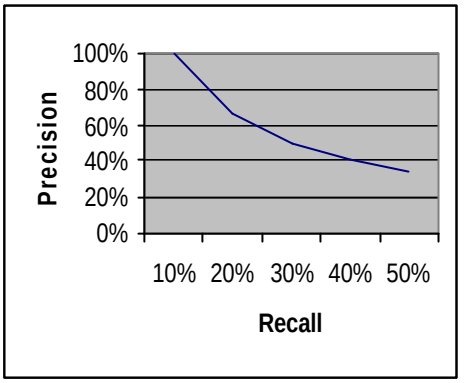


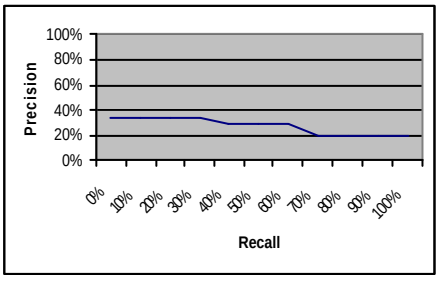
Gráfico Precision Recall. Interpolación

Relev	3
Recs	15

	nr	nr	r	nr	nr	nr	r	nr	nr	nr	nr	nr	nr	nr	r
Doc Rec	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rel Rec	0	0	1	1	1	1	2	2	2	2	2	2	2	2	3

Recall	0%	0%	33%	33%	33%	33%	67%	67%	67%	67%	67%	67%	67%	67%	100%
Precision	0%	0%	33%	25%	20%	17%	29%	25%	22%	20%	18%	17%	15%	14%	20%

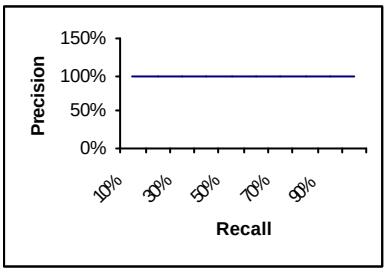
Gráfico			
	33%	67%	100%
	33%	29%	20%



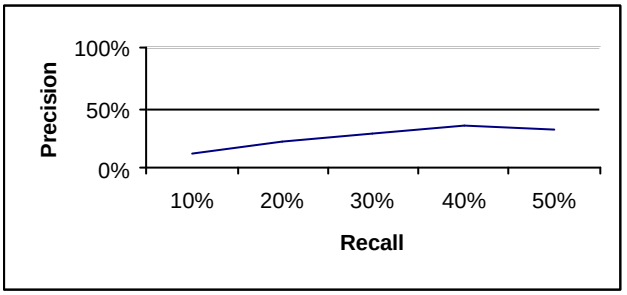
Interpolación

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
33%	33%	33%	33%	29%	29%	29%	20%	20%	20%	20%

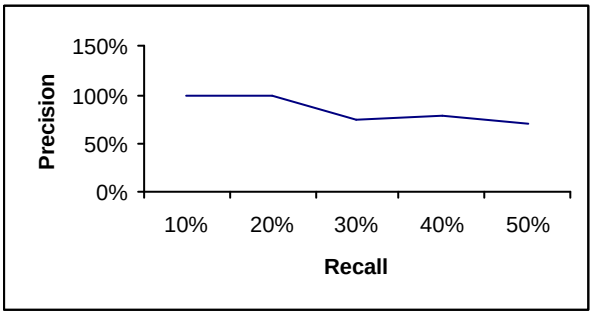
Gráficos de precisión recall



Recuperación idónea
Cada documento recuperado es relevante



Recuperación tardía
Los primeros docs no son relevantes
pero los últimos si



Recuperación temprana
Los primeros docs son relevantes
pero los últimos no

Consultas agrupadas

- Los gráficos de precision recall no suelen contener una sola consulta, sino que agrupan varias consultas
- El método es calcular la precision media a cada uno de los 11 niveles de recall

Estimación Recuperación en Internet

- Problema:
  - Se desconoce el total de relevantes (Recall)
  - Difícil conocer el total de relevantes recuperados si la búsqueda tiene muchos docs
  - Dificultades añadidas por documentos no indizados por el motor y documentos no recuperados pero indizados por el motor
  - Para poder comparar motores en Internet deberíamos de poder utilizar la BD de un motor (p.e. Google) con los algoritmos de recuperación y posicionamiento de otro motor (p.e. Altavista)

Estimación Recuperación en Internet: Soluciones

- No calcular la Recall
- Limitarse a los n primeros resultados recuperados (20)
- Utilizar palabras de muy baja presencia para así poder evaluar todos los documentos
- Para Comparar motores: A veces se normaliza el número total de relevantes sumando los documentos relevantes de los 20 primeros resultados de varios motores
- Identificar documentos que deberían de estar (p.e.por estar en una revista electrónica o un dominio relevante), ver cuantos recupera
- Poner artículos relevantes en el motor y ver cuantos se recuperan
- Si se puede acceder a subcolecciones como newsgroups hacer muestreos de relevantes

Estimación Recuperación en Internet

- Algunos autores (Chignell) proponen modificar la medida de Precision de los 20 primeros resultados añadiendo información sobre el grado de Relevancia

P=Spuntuación/20*4

La puntuación se asigna manualmente de 1 (mínimo) a 4 (máximo)

Consultas sin Agrupar

- **Desventajas de Agrupar**
  - **No se puede saber como se comporta un tipo específico de consultas**
  - **No permite comparar dos algoritmos frente a consultas individuales**
    - **Tipos:**
      - **Media de Precision en n valores de recuperación**
      - **R-Precision**
      - **Histogramas de Precision**

Consultas sin agrupar

Relevantes	10
Recuperados	15

R-Precision = 40%

Valor de la precisión al recuperar el mismo nºde docs q el nº de documentos relevantes

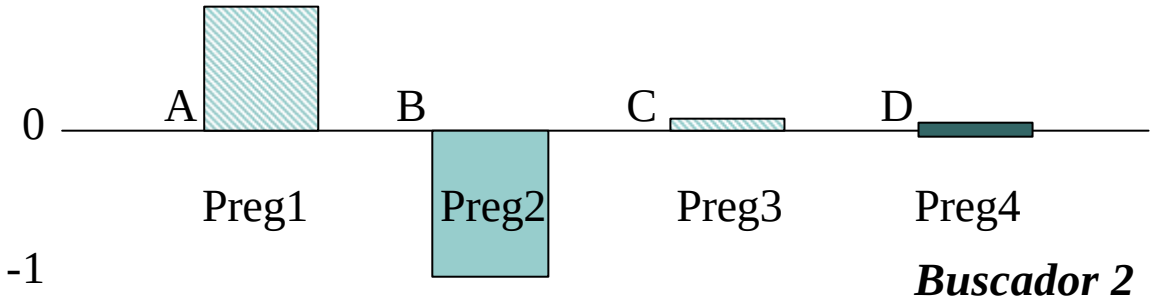
Documentos Recuperados	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Relevantes Recuperados	1	1	2	2	2	3	3	3	3	4	4	4	4	4	5
Recall	10%	10%	20%	20%	20%	30%	30%	30%	30%	40%	40%	40%	40%	40%	50%
Precision	100%	50%	67%	50%	40%	50%	43%	38%	33%	40%	36%	33%	31%	29%	33%

Precisión media a n documentos relevantes

10%	20%	30%	40%	50%
100%	67%	50%	40%	33%

=suma porcentajes dividido número de relevantes recuperados **58%**

Consultas sin agrupar: Histogramas de Precision



Buscador 2

- Se representa R-precision de cada consulta en 2 buscadores distintos
- Se resta el valor de la R-precision en el buscador 1 al de la R-precision en el buscador 2

A-Buscador 1 mejor que el 2 en la primera pregunta
B-Buscador 2 mejor en la segunda pregunta
C-Buscador 1 un poco mejor
D-Los dos buscadores son idénticos

Medidas orientadas al usuario

Para un usuario concreto	Conocidos	Desconocidos
Relevantes Recuperados	A	B
Relevantes (presentes o no en la BD)	C	D

- Cobertura=A/C

De los relevantes conocidos por el usuario cuantos se han recuperado

- Novedad=B/(A+B)

De los relevantes recuperados cuantos le eran desconocidos

Medidas Centradas en el Usuario

- Recall Relativa:
  Documentos relevantes recuperados/Documentos relevantes esperados

- Esfuerzo en la Recuperación:
  Documentos relevantes esperados/Documentos relevantes examinados

Colecciones de Prueba: Test collections

- Las tasas de Precision Recall son solo ciertas para determinada colección y determinadas preguntas, no es extrapolable
- Colecciones predefinidas de documentos, preguntas y juicios de relevancia (ajuste de cada documento a cada pregunta)→Benchmarking
- Sirven para mejorar los algoritmos de recuperación y posicionamiento
- Tendencia a ajustarse a la realidad. En sus inicios eran documentos breves y las preguntas no eran las típicas de los usuarios
- En un principio con etiquetas propias, actualmente con DTDs de XML
- Existen competiciones en que varios motores muestran sus prestaciones:
  - TREC (Recuperación), Message Understanding Conferences (MUC), Document Understanding Conferences (DUC), Cross-Language Evaluation Forum (CLEF), Summarization evaluation effort (SUMMAC), SENSEVAL (Semántica), CLEF (Multilingüe). Los documentos en TREC están marcados en html.
  - Colecciones clásicas: ftp://ftp.cs.cornell.edu/pub/smart

Colecciones clásicas (Smart)

COLECCIÓN		DOCS	terms	PREG	terms	TAMAÑO	
CACM	Informatica	3,204	10,446	64	11,4	1.5	
CISI	Biblio.	1,460	7,392	112	8,1	1.3	
CRAN	Aeronau.	1,400	258,771		225	4043	1.6
MED	Medicina	1,033		30		1.1	
TIME	Articulos	425		83		1.5	

*Tenían los documentos muchas preguntas y por lo general eran muy breves.

Cranfield

- Ejemplo documento

.I 250
.T pressure distributions at zero lift for delta wings with rhombic cross sections .
.A eminton,e.
.B arc cp.525, 1960.
.W pressure distributions at zero lift for delta wings with rhombic cross sections ... calculation and some of the results are compared with those of slender thin wing theory .

- Ejemplo pregunta

.I 029
.W material properties of photoelastic materials .

Cranfield

- Evaluación

Pregunta ID	Documento ID	Grado Relevancia
29	225	3
29	250	2
29	464	4
29	513	-1

Campos en las colecciones clásicas
Con estos campos se hacía la recuperación:

- Título, Autor, Fuente (casi todas)
- Resumen (Cranfield, CISI, Time, Medline)
- Fecha (Time,CACM)
- Raíces de palabras (CACM, CISI)
- Referencias (CACM)
- Categoria (CACM)
- Cocitaciones (CACM, CISI)

- Preguntas con autor y su perfil de trabajo (CACM)
- Glosario (Time, CACM)

TREC

- Antiguo TIPSTER, organizado por NIST y por DARPA
- Existen distintas modalidades, algunos son:
    - Ad hoc: Aparecen nuevas preguntas pero el corpus de documentos es fijo
    - Routing: Aparecen nuevos documentos pero el corpus de preguntas es fijo. Existe un corpus de entrenamiento
    - Grandes Corpus: de hasta 8 millones de documentos
- TREC tiene estadísticas propias de análisis que son las que la han dado su aceptación

Ejemplo Documento

<DOC>
<DOCNO> WSJ870324-0001 </DOCNO>
<HL> John Blair Is Near Accord To Sell Unit, Sources Say </HL>
<DD> 03/24/87</DD>
<SO> WALL STREET JOURNAL (J) </SO>
<IN> REL TENDER OFFERS, MERGERS, ACQUISITIONS (TNM) MARKETING, ADVERTISING (MKT) TELECOMMUNICATIONS, BROADCASTING, TELEPHONE, TELEGRAPH (TEL) </IN>
<DATELINE> NEW YORK </DATELINE>
<TEXT>
John Blair & Co. is close to an agreement to sell its TV station advertising representation operation and program production unit to an investor group led by James H. Rosenfield, a former CBS Inc. executive, industry sources said. Industry sources put the value of the proposed acquisition at more than $100 million. ...
</TEXT>
</DOC>

TREC Consulta

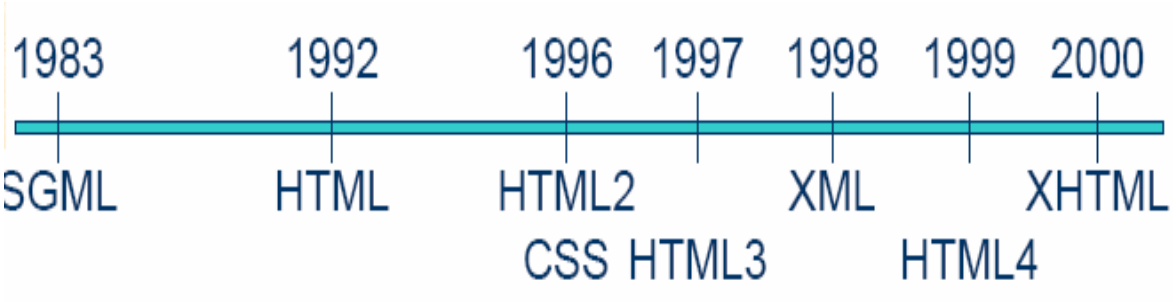
<top> <head> Tipster Topic Description
<num> Number: 066
<dom> Domain: Science and Technology
<title> Topic: Natural Language Processing
<desc> Description: Document will identify a type of natural language processing technology which is being developed or marketed in the U.S.
<narr> Narrative: A relevant document will identify a company or institution developing or marketing a natural language processing technology, identify the technology, and identify one of more features of the company's product.
<con> Concept(s): 1. natural language processing ;2. translation, language, dictionary
<fac> Factor(s): <nat> Nationality: U.S.</nat></fac>
<def> Definitions(s): </top>

Tema 3: XML: eXtensible Markup Language

(no entra ni Schemas XML Path)

Introducción Histórica (I)

- XML se constituyó como estándar de la W3C en el año 1998. En 2000 se aprueba su versión 1.0
- Se trata de un lenguaje de marcas, igual que HTML o su precursor SGML
- Se diferencia de SGML por su sencillez
- Se diferencia de HTML por su flexibilidad: el número de etiquetas que puede incluir un documento XML es ilimitado
- Al igual que HTML, es portable a cualquier plataforma



- **Objetivos principales:**

– Directamente utilizable en Internet
– Soporte para una amplia variedad de aplicaciones para transferencia de datos
– Compatible con SGML
– Posibilidad de crear sencillos procesadores de XML
– Documentos XML legibles y medianamente claros (depende de la definición)
– Diseño rápido del lenguaje
– Simple, pero perfectamente formalizado
– Documentos XML fáciles de crear

XML vs. HTML

- HTML carece de un chequeo sintáctico. Páginas con errores son mostradas en los navegadores
- HTML carece de estructura
- HTML no es orientado a objeto
- HTML mezcla contenido y representación
- Por todo esto:

– HTML no puede ser fácilmente leído por una máquina
– HTML nunca será un estándar de intercambio de datos

- XML cubre todo esto con un lenguaje de sencillez extrema

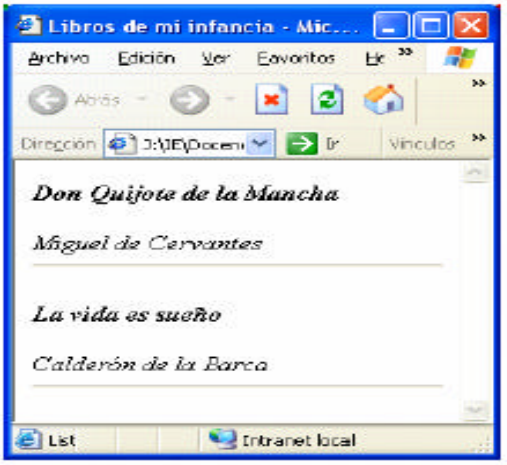
Características de XML (I)

- Es un subconjunto del lenguaje SGML
- Al igual que él, se utiliza para representar datos de forma estructurada (Jerárquica)

- Se basa en una gramática de obligado cumplimiento. Esto facilita el desarrollo de *parsers* y, por lo tanto, su utilización masiva
- La estructura interna de un documento XML puede reflejarse en otro documento denominado DTD (*Document Type Definition*)
- A diferencia de HTML, separa radicalmente la semántica del documento, de su representación gráfica

Documento HTML (I)

```
<HTML>
<HEAD><TITLE>Libros de mi
   infancia</TITLE></HEAD>
<BODY>
<P><I><B>Don Quijote de la
   Mancha</B>
<P><I>Miguel de Cervantes</I>
<HR>
<P><B>La vida es sueño</B>
<P><I>Calderón de la Barca</I>
<HR>
</BODY>
</HTML>
```



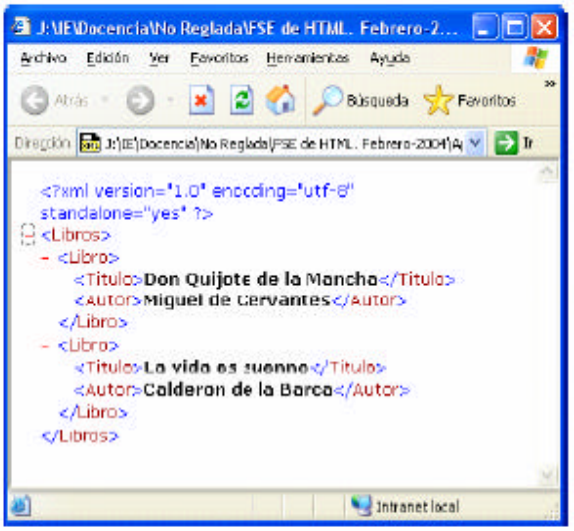
- En apariencia, el documento HTML anterior es correcto, sin embargo:

– Existen etiquetas que nunca se cierran: <P>
– Algunas etiquetas no están bien anidadas: el primer <I> nunca se cierra
– Para un lector no humano, no se sabe qué es un libro y qué es un autor

- XML erradica todos estos problemas!!

Documento XML

```
<?xml version="1.0" encoding="utf-8"
    standalone="yes" ?>
<Libros>
    <Libro>
        <Titulo>Don Quijote de la
    Mancha</Titulo>
        <Autor>Miguel de
    Cervantes</Autor>
    </Libro>
    <Libro>
        <Titulo>La vida es suenno</Titulo>
        <Autor>Calderon de la
    Barca</Autor>
    </Libro>
</Libros>
```



Reglas Generales de XML

- Un único elemento raíz
- Todo elemento debe tener etiquetas de apertura y cierre
- Distinción entre mayúsculas/minúsculas
- Anidamiento perfecto entre elementos
- Los valores de atributos siempre van entre comillas
- Los espacios en blanco se conservan
- Los caracteres CR/LF se transforman en LF

Documentos Bien Formados y Válidos

- Se dice que un documento es bien formado cuando:

– Cumple con todas las reglas anteriormente expuestas
– Contiene uno o más elementos
– Hay un único elemento raíz (elemento documento)
– Si el documento consta de más de una parte, todas están bien formadas
– No se encuentran caracteres prohibidos en el texto

- Un documento es válido cuando, además de ser 'bien formado', cumple con las especificaciones semánticas expuestas en su plantilla (DTD o XML Schema)

Elementos

- Comentarios:

– <! -- Esto es un comentario, y no puedo incluir un doble guión-->

- Instrucciones de procesamiento:

– <? Instrucción ?>
– La instrucción no puede incluir los caracteres ?>

- Secciones CDATA:

– <![CDATA[Este texto no será tratado, puede incluir "cualquier" &carácter < >]]>
– No son tratadas por el *parser*
– Pueden incluir cualquier carácter prohibido (", ', &, >, <).
– No puede incluir la cadena]]>

- Prólogo:

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
– Es una instrucción de procesamiento obligatoria
– Version: indica la versión de XML que se está utilizando (1.0 en la actualidad). *Es obligatoria*
– Encoding: indica cómo se codificó el documento, y *no es obligatoria* (por defecto UTF-8). Válido para otros juegos de caracteres
– Standalone: "yes" indica que el documento no va acompañado de DTDs externos; "no" indica que posee DTD interno. *No es un atributo obligatorio*

- DOCTYPE: **<!DOCTYPE MiDTD SYSTEM "C:\MiDTD.dtd">**

– Indica la referencia (URI) al DTD, así como el nombre (MiDTD) del elemento raíz de la misma
– La DTD podría ir incorporada en el propio documento XML, sin requerir otro fichero aparte
– El documento XML deberá cumplir con el contenido del DTD

- Etiquetas:

– Deben ir correctamente anidadas: apertura y cierre
– Etiqueta de apertura: comienza por <, más el nombre de la etiqueta y terminan por >. Ejemplo <Libro>
– Etiqueta de cierre: </Libro>
– Etiqueta vacía: <Libro />
– No puede iniciar el nombre con ".", ":", "-", números
– Luego de la primera letra pueden colocarse ".", números, "-"
– El nombre debe comenzar por una letra o un "_"
– No puede comenzar por "xml"

- Elemento:

– Es el conjunto de la etiqueta (marcador) de apertura, su contenido y la de cierre
– Por ejemplo: <Libro>Don Quijote de la Mancha</Libro>
– Hay algunos caracteres reservados (prohibidos):
  - Signo de mayor: >
  - Signo de menor: <
  - Ampersand: &
  - Apóstrofe: '
  - Comilla: "

– Estos caracteres prohibidos se reemplazan por entidades o se incluyen en secciones CDATA

- Atributos:

– Cada elemento puede contener 0 ó más atributos

– Su valor debe ir siempre entrecomillado
– Sólo pueden aparecer en etiquetas de apertura o vacías
– El mismo atributo no puede aparecer repetido en la misma etiqueta
– Si el documento incluye DTD, cada atributo debe estar definido como atributo del presente elemento
– No puede contener ninguna referencia a entidad externa
– Son siempre tratados como cadenas de texto

```
<Libro>
<Titulo>Don Quijote de la Mancha</Titulo>
<Autor>Miguel de Cervantes</Autor>
<Precio> 1.123 euros </Precio>
<Editorial> Santillana </Editorial>
</Libro>

<Libro Precio = "1.123 euros" Editorial = "Santillana">
<Titulo>Don Quijote de la Mancha</Titulo>
<Autor>Miguel de Cervantes</Autor>
</Libro>
```

(Sin atributos)

(Dos elementos son atributos)

DTDs (I) (Declaración de tipos)

Externa

```
<!DOCTYPE Libros SYSTEM "Libros1.dtd">
<Libros>
        <Libro>
        <Titulo>Don Quijote de la Mancha</Titulo>
        <Autor>Miguel de Cervantes</Autor>
        </Libro>
        <Libro>
        <Titulo>La vida es sueño</Titulo>
        <Autor>Calderon de la Barca</Autor>
        </Libro>
        </Libros>
```

Interna

```
<!DOCTYPE Libros [
<!ELEMENT Libros (Libro)+>
<!ELEMENT Libro (Titulo, Autor)>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Autor (#PCDATA)>
]>
<Libros>
<Libro>
<Titulo>Don Quijote de la Mancha</Titulo>
<Autor>Miguel de Cervantes</Autor>
</Libro>
<Libro>
<Titulo>La vida es suenno</Titulo>
<Autor>Calderon de la Barca</Autor>
```

DTDs (II)

- Toda DTD debe tener uno y sólo un elemento raíz (también conocido como elemento documento)
- Este documento raíz debe coincidir con el nombre que aparece a continuación del DOCTYPE
- Un documento DTD puede contener:

– Declaraciones de elementos
– Declaraciones de atributos para un elemento
– Declaraciones de entidades
– Declaraciones de notaciones
– Instrucciones de procesamiento
– Comentarios
– Referencias a entidades de parámetro

DTDs (III) (Elemento Raíz)

- A partir del elemento raíz, pueden opcionalmente colgar (de forma jerárquica) otros elementos

<!ELEMENT Libros (Libro)+>
<!ELEMENT Libro (Titulo, Autor)>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Autor (#PCDATA)>

DTDs (IV) (Contenido de los Elementos)

- Contenido de un elemento:

– EMPTY: el elemento está vacío (puede contener atributos).
<!ELEMENT IMAGEN EMPTY>
– ANY: el elemento puede contener a cualquier otro elemento o incluso contenido textual.
<!ELEMENT IMAGEN ANY>
– Otros elementos: un elemento puede contener uno o más elementos hijos en una cierta secuencia (Ej. Libro)
– #PCDATA: texto *parseado.*
<!ELEMENT LIBRO (#PCDATA)>
– Mixto: el elemento puede incluir secuencias de caracteres opcionalmente mezcladas con elementos hijos.
<!ELEMENT LIBRO (#PCDATA | AUTOR)>*

DTDs (V)

- Secuencias de hijos de un elemento:

– Secuencia:
Secuencia en orden: hijos separados por comas
Opciones: hijos separados por | (barra)
Conjuntos de elementos pueden agruparse entre paréntesis
– Cardinalidad: un elemento, o un conjunto de ellos
puede repetirse 0, 1 ó más veces:
elemento Elemento repetido 1 única vez

?	Elemento repetido 0 ó 1 vez
*	Elemento repetido 0 ó más veces
+	Elemento repetido 1 ó más veces

DTDs (VI)

DTDs (VII) (Ejemplo)

<!ELEMENT LIBRO (Autor, Editorial)>
<!ELEMENT Autor (#PCDATA)>
<!ELEMENT PELICULA (Actor|Actriz|Director)+>
<!ELEMENT PELICULA ((Actor | Actriz)*, Director, Maquillaje?)>
<!ELEMENT PELICULA (#PCDATA | Actor)*>
<!ELEMENT PELICULA (Titulo, Genero, (Actor | Actriz | Narrador)*)>
<!ELEMENT FICHA (Nombre+, Apellido+, Direccion*, foto?,
TelFijo*|TelMovil*)

Ejercicio: Hacer una DTD.

```
<?xml version="1.0" encoding="utf-8" ?>
<Agenda>
<Persona>
    <Nombre> Anabel </Nombre>
    <Apellido> Fraga </Apellido>
    <Email> afraga@ie.inf.uc3m.es </Email>
    <Oficina>  2.1 B18 </Oficina>
    <Telefono> 5555555 </Telefono>
    <Movil> 5557777 </Movil>
</Persona>
</Agenda>
```

DTDs (IX) (Atributos)

- Un elemento puede opcionalmente declarar uno o más atributos

<!ATTLIST Elemento Atributo Tipo Modificador>

- Los atributos de un elemento pueden incluirse en una o más declaraciones <!ATTLIST ...>. Si se hace en la misma declaración, basta con separar con un espacio (espacio, tabulador, retorno de carro)
- Tipo de un atributo:

– Tipo cadena: CDATA

<!ATTLIST Autor Nacionalidad CDATA>

– Tipo enumerado:

<!ATTLIST Pelicula Genero (Ficcion | Terror | Humor)>

– Tipo simbólico:

  - **ID**: valdrá como identificador en el resto del documento, sólo un atributo ID por cada elemento
  - **IDREF**, **IDREFS**: su valor debe coincidir con algún otro atributo de tipo ID en el resto del documento XML. IDREFS separa las referencias por espacio. "ID1 ID2 ID3"
  - **ENTITY**, **ENTITIES**: su valor debe coincidir con una o más entidades no analizadas
  - **NMTOKEN**, **NMTOKENS**: su valor ha de ser una cadena de tipo *token*

DTDs (XI) (Modificadores de Atributos)

- Modificadores:

– #REQUIRED: este atributo debe ser obligatoriamente introducido.

<!ATTLIST Pelicula Titulo CDATA #REQUIRED>

– #IMPLIED: indica que el atributo es *opcional*

– ValorPredeterminado: si se omitiese el atributo, los procesadores recogerían este valor por defecto

<!ATTLIST Pelicula Genero (Ficcion | Terror | Humor) "Humor">

<!ATTLIST Autor Nacionalidad CDATA "Española">

– #FIXED: se incluya o no se incluya el atributo, los procesadores siempre obtendrán este mismo valor

<!ATTLIST Autor Nacionalidad CDATA #FIXED "Espanola">

DTDs (XII) (Recomendaciones para modelado de Atributos)

- Frecuentemente un mismo objeto se puede diseñar como un atributo o un elemento sin pérdida de semántica pero existen criterios para decantar...
- Atributos
  - Normalmente se trata de objetos cuya existencia no tiene sentido fuera del objeto al que describen (p.e.adjetivos), metadatos, identificadores únicos, el idioma, ...
  - En general, todo aquello por lo que existe mayor interés en filtrarlo que en mostrarlo

    Ventajas
    - Más fácil de procesar por el software (mayor eficiencia)
    - Más legible, los atributos están próximos al elemento al que pertenecen

- Elementos, se debería optar por ellos...
    - Siempre que se quiera definir sub-elementos (ya sean partes del elemento principal o sus específicos)
    - Permiten crear vínculos
    - Siempre que queramos repetir el mismo elemento con distinto valor (los atributos tienen un único valor como máximo)
    - Siempre que el contenido sea mayor que una palabra (oraciones, párrafos, ...) y sobre todo si se quiere mostrar el texto en cuestión
    - Tienen entidad propia independientemente del resto de elementos

Los documentos que priman a los atributos pueden ser más breves al no tener una etiqueta propia que abrir y cerrar

DTDs (XIV) (Problemas)

- Una DTD no sigue el formato de un documento XML estándar. Esto representa un problema para los *parsers*
- No se soportan distintos tipos de datos al estilo de los lenguajes de programación (CDATA, #PCDATA)
- No se pueden crear tipos de datos personalizados
- No se soportan los espacios de nombres (namespaces)
- El número de ocurrencias no se puede controlar al 100% (Ej. 2 ocurrencias)
- Por estas y otras razones, surgen los XML *Schemas* (Esquemas)

Referencias a Caracteres

- Permiten incluir cualquier carácter dentro de un documento XML
- Basado en el conjunto de caracteres ISO/IEC 10646 (http://xml.coverpages.org/xml-ISOents.txt)
- Dos formatos:

– &#x*valor*; valor representado en decimal
– &#x*valor*; valor representado en hexadecimal

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Libros>
<Libro Precio = "658 euros" Editorial = "Anaya">
<Titulo>La vida es sue&#241;o</Titulo>          (DECIMAL)
<Autor>Calder&#243;n de la Barca</Autor>   (DECIMAL)
</Libro>
</Libros>
```

Entidades (I)

- Las entidades permiten:

– Dar modularidad al texto evitando tener que escribir algo de forma repetitiva (Re-uso)
– Incluir caracteres prohibidos &, >, <, “, ‘
– Incluir caracteres de otros idiomas: eñe, ...

- Comienzan por & y terminan en “;” Como por ejemplo: *&*

Entidades (II)

- Entidades predefinidas:

– Signo menor lt < <
– Signo mayor gt > >
– Ampersand amp & &
– Apóstrofe apos ‘ '
– Comilla doble quot “ "

Entidades (III)

- Tipos de entidades:

– General y de Parámetro:
General: contiene texto XML u otros caracteres
De Parámetro: contiene texto XML que puede insertarse dentro de una DTD
– Interna y Externa:
Interna: contiene el texto dentro de una cadena entrecomillada
Externa: hace referencia a un archivo externo
– Analizada y no Analizada:
Analizada: texto XML que será *parseado* en su punto de inserción
No Analizada: no será *parseada*

Ejercicio de Atributos:

- Hacer una DTD utilizando atributos:

```
<?xml version="1.0" encoding="utf-8" ?>
<Agenda>
<Persona>
    <Nombre> Anabel </Nombre>
    <Apellido> Fraga </Apellido>
    <Sexo> Femenino </Sexo>
    <DNI> 44444444-O </DNI>
    <Nacionalidad> Española </Nacionalidad>
    <Email> anabel_fraga@mydomain.es </Email>
 <Email> anabel_fraga@fraga.es </Email>
    <Oficina>
                    <Direccion  CP=”28911”> Av. Universidad  30 </Direccion>
                    <Despacho> 2.1 B18  </Despacho>
                    <Email> afraga@ie.inf.uc3m.es </Email>
                    <Telefono> 5555555 </Telefono>
                    <Telefono> 5555556 </Telefono>
 </Oficina>
 <Telefono> 5555558 </Telefono>
 <Telefono> 5555559 </Telefono>
    <Movil> 5557777 </Movil>
</Persona>
</Agenda>
```

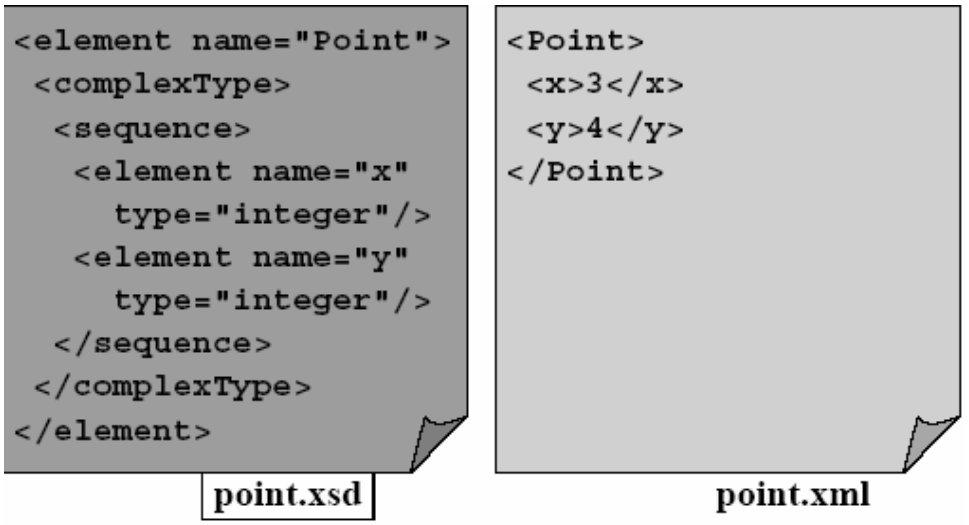
XML Schemas (I)

- Actualmente existe una nueva recomendación de W3C de Mayo 2001 para definiciones de XML:

XML Schemas

- Uso de notación XML para definiciones
- Limitación de uso: Actualmente existe una gran cantidad de documentos definidos con DTDs.

XML Schemas (II) (Ejemplo)



XML Schemas vs. DTDs (I)

Desventajas de las **DTDs**

- No escritas en sintaxis XML
- Poco uso de namespaces
- Pocos tipos de datos (y lo que es peor, no se pueden definir nuevos tipos de datos)
- Aunque se puede agrupar elementos mediante entities (%;) están poco desarrolladas

Ventajas de las **DTDs**

- Muchas herramientas que lo soportan
- Existen muchos documentos: DTDs y XMLs basados en ellas
- Fácil de aprender

XML Schemas vs. DTDs (II)

- Ventajas:
  - Permite multitud de tipos de datos (pe xs:date, xs:int, xs:language, ...)
  - Amplio uso de los namespaces
  - Permite agrupar elementos para su reutilización, permite herencia (Ejemplo: Datos Personales en distintos Dominios de uso)

La Familia XML (I)

– XPointer/XLink: permiten referenciar a diferentes recursos, dentro o fuera del documento XML

- XPath: lenguaje de consulta para recorrer ficheros XML
- XQL (XML Query Language): útil para localizar y extraer elementos de un documento XML
- XIRQL: Una extensión de XQL para Recuperación de Información
- XSLT: Lenguaje para transformación de documentos XML
- XSL-FO: Expresa semántica de formateado de documentos, provee los medios para producir impresiones de alta calidad.

XPath (II) (Ejemplo) no entra en el examen

```
<catalogo>
<libro>
<titulo>Professional XML</titulo>
<autor>Didier Martin et al.</autor>
<editorial>Wrox</editorial>
<anyo>2000</anyo>
</libro>
<libro>
<titulo>XML Developer's Guide</titulo>
<autor>Fabio Arciniegas</autor>
<editorial>McGraw-Hill</editorial>
<anyo>2001</anyo>
</libro>
</catalogo>
```

• Todos los autores:
"/catalogo/libro/autor"
"/catalogo/*/autor"
"//autor"
• Todos los autores, con condición:
"/catalogo/libro[anyo>2001]/autor"
• El texto de los elementos autor:
"/catalogo/libro/autor/text()"
• El primer libro:
"/catalogo/libro[0]"

XPath (III)

- Expresiones numéricas
  + - * div mod
- Expresiones booleanas
  and or
- Expresiones de comparación
  = != < <= > >=

XPath (IV)

- Funciones numéricas

– round ceiling floor
– count number sum

- Funciones booleanas

– boolean false true not

- Funciones de cadenas de caracteres

– string string-length substring
– substring-after substring-before
– contains starts-with concat
– normalize translate

XPath (V) (Unión)

- "|" sirve para calcular la unión de conjunto de nodos especificados por medio de *location paths*
- Ejemplos:

"//libro[anyo=2000]|//libro[anyo=2001]"
"//libro[anyo=2000 or anyo=2001]"

Presentación en XML

- La presentación en HTML esta básicamente en los navegadores.
- Sería interesante *programar* la presentación (re-uso de código)
- Surgen las hojas de estilo:

– **CSS**: Cascading Style Sheets (HTML)
– **XSL**: eXtensible Style Language (XML) (XML + DTD o XML Schema + Fichero de Estilo XSL)

XSL

- Xsl es el lenguaje basado en xpath para poner hojas de estilos en xml, es decir sustituye a las css
- Para hacer referencia a un fichero xsl basta con escribir en el documento xml tras el prólogo la frase:

<?xml-stylesheet type="text/xsl" href="nombrefichero.xsl" ?>

Namespaces (I)

- XML permite crear etiquetas 'casi' sin ninguna limitación en sus nombres
- Esto implica que, mezclar dos documentos, con diferentes etiquetas, podría resultar en una duplicidad de etiquetas
- Mediante la definición de espacios de nombres, se pueden evitar estas colisiones
- Tecnologías como XSL y otras muchas hacen uso de *Namespaces*

Namespaces (II) (Definición)

- Un *namespace* se identifica por su prefijo.

Por ejemplo:
<xsl:stylesheet ***xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"***>
donde:
– **xsl** es el prefijo del *namespace*
– **Stylesheet** es el nombre completo del *namespace*
– **http://www...** es la URI donde se puede encontrar más información sobre el estándar
– Puede incluir otros atributos como ***version***...

– Como todo elemento XML, ha de **cerrarse** *</xsl:stylesheet>*

Namespaces y qualified names (Qnames)

- Para escribir en XML metadatos hay que definir previamente la ubicación del vocabulario de metadatos (Namespaces) y un prefijo para hacer referencia al vocabulario empleado (Qname)

Ejemplo NameSpace

- <?xml version="1.0"?>
  <!-- initially, the default namespace is "books" -->
  <book xmlns='urn:loc.gov:books' xmlns:dc="dublincore.org"
  xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>XML el futuro</title> <dc:creator>yo mismo</dc:creatro>
  <isbn:number>1568491379</isbn:number>
  <notes>
  <!-- esto es un comentario en el ejemplo y html -->
  This is a funny book!
  </notes>
  </book>

Metadatos

- Registros: repositorios para gestionar, recuperar, referenciar y reutilizar vocabularios de metadatos existentes. Estos registros suelen facilitar información sobre la definición, origen y localización del recurso. Actualmente, estándar ISO 11179
  - Proyecto Schemas, para RDF(S) y namespaces relacionados con proyectos de la UE (http://www.schemas-forum.org/);
  - Open Metadata Registry y ULIS son otros proyectos que recopilan metadatos relacionados con la Dublín Core Metadata Iniciative (http://dublincore.org/dcregistry/navigateServlet y http://avalon.ulis.ac.jp/registry/);
- Perfiles de Aplicación

Metadatos famosos

- DC (Dublín Core) para la descripción de documentos. Con cualificadores
- FOAF (Friend of a Friend) vocabulario sobre información personal y relaciones interpersonales. Sin vocabulario estable para realizar extensiones. Es generado automáticamente en websites que trabajan con blogs.
- RSS (RDF Site Summary una de las siglas hay pa gustos) tiene un conjunto de metadatos multipropósito, suele ser utilizado principalmente para describir sitios web. Sindicate pages
- Text Encoding Initiative (TEI) (http://www.tei-c.org/) marcado de e-text.
- Encoded Archival Description (EAD) para descripción de archivos y colecciones especiales.

Según Swoogle en Junio de 2004 que los NS, asociados a vocabularios de metadatos, más utilizados eran: FOAF (1.126.002 documentos), DC (1.126.002), MCVB (8.838), RSS(7.560 Junio, 80.000 septiembre de 2004), vCard (6.229) y Bio (6.183).

Etiqueta META en HTML

La etiqueta META se utiliza **dentro del encabezamiento HEAD** de una página HTML, para identificar, indizar y catalogar documentos

Los atributos de esta etiqueta se encuentran indicados en el RFC 1866 bajo la siguiente DTD (Document Type Definition):

```
<!ELEMENT META>
<!ATTLIST META
  http-equiv  NAME        #IMPLIED
  name        NAME        #IMPLIED
  content     CDATA       #REQUIRED >
```

Editores metadatos

- http://www.ukoln.ac.uk/cgi-bin/dcdot.pl
- http://vancouver-webpages.com/META/mk-metas.html
- http://www.lub.lu.se/cgi-bin/nmdc.pl
- Reggie metadata.net/dstc
- http:www.ukoln.ac.uk/metadata/new-dcdot
- http://rainbow.arch.scriptmania.com/tools/adv_me tatag_generator.html

Dublin Core: elementos (1)

Los 13 elementos iniciales:

- Subject
- Title
- Author
- Publisher
- OtherAgent
- Date
- ObjectType
- Form
- Identifier
- Relation
- Source
- Language
- Coverage

Ejemplo de un documento DC HTML

```
<META NAME="Title" CONTENT="FrontOffice selects Verity for Microsoft Exchange basad
document management system">
<META NAME=“DC.Author" CONTENT="Padovani, Marguerite">
<META NAME=“DC.Author" CONTENT="Siegel, Gail">
<META NAME=" DC.Publisher" CONTENT="Verity Inc.">
<META NAME=" DC.Date" CONTENT="1996">
<META NAME=" DC.Object" CONTENT="Press Release">
<META NAME=" DC.Form" CONTENT="1 ASCII file">
<META NAME=" DC.Language" CONTENT="English">
```

Calificadores DC (1)

- Propuestos en DC4 como Canberra Qualifiers.
  - Esquema.nombre_de_elemento.nombre_de_sub-elemento = “valor”
  - DC.Creator.personalName =”Scott Adams”

- Aprobación, el 17-04-00, de Dublin Core Qualifiers (qDC)
  - Lista no cerrada que formaliza el método de utilización

Elemento DCMES	Elemento refinado	Sistema de codificación
Title	Alternative	
Creator		
Subject		LCSH MeSH DDC LCC UDC
Description	Table Of Contents Abstract	
Publisher		
Contributor		
Date	Created Valid Available Issued Modified	DCMI Period W3C-DTF
Type		DCMI Type Vocabulary
Format	Extent	
	Medium	IMT
Identifier		URI
Source		URI
Language		ISO 639-2 RFC 1766
Relation	Is Version Of Has Version Is Replaced By Replaces Is Required By Requires Is Part Of Has Part Is Referenced By References	URI

	Is Format Of Has Format	
Coverage	Spatial	DCMI Point ISO 3166 DCMI Box TGN
	Temporal	DCMI Period W3C-DTF
Rights		

Interoperatibilidad

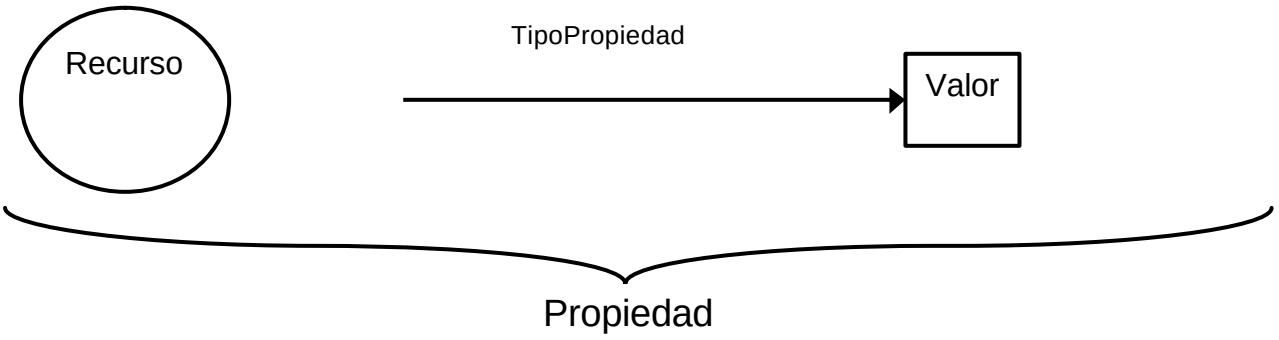
- El término interoperabilidad ha sido definido (ALA, 2000) como la capacidad que tienen algunos sistemas para intercambiar y utilizar información procedente de otro sistema diferente.
- La forma más usual de que exista es alineando vocabularios de metadatos y esquemas XML.
- La alineación puede ser estructural o lingüística

RDF (Resource Description Framework)

Validadores: http://zoe.mathematik.uni-osnabrueck.de/RDF/parser.html
http://www.w3.org/RDF/Validator/
Objetivos de RDF:

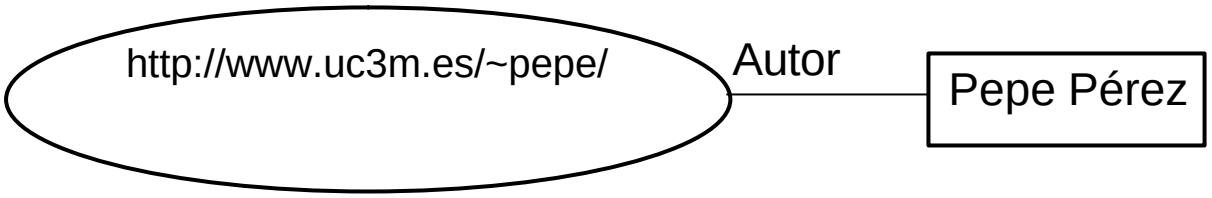
- interoperabilidad de metadatos a través de diferentes descripciones de recursos Web e intercambio de Metadatos.
- RDF trata de hacer compatibles diferentes estándares.
- Marco genérico de descripción de recursos
  - Colección de propiedades=RDF.
  - Cada propiedad tiene un tipo de propiedad y un valor
- Formato de metadatos
- Interoperabilidad entre aplicaciones
- Intercambio de descripciones de recursos

El modelo RDF



- Basado en un modelo matemático=triple
- Recursos Web representados por nodos URI
- Los conjuntos de propiedades se conocen como “descripciones”

RDF – ejemplo básico

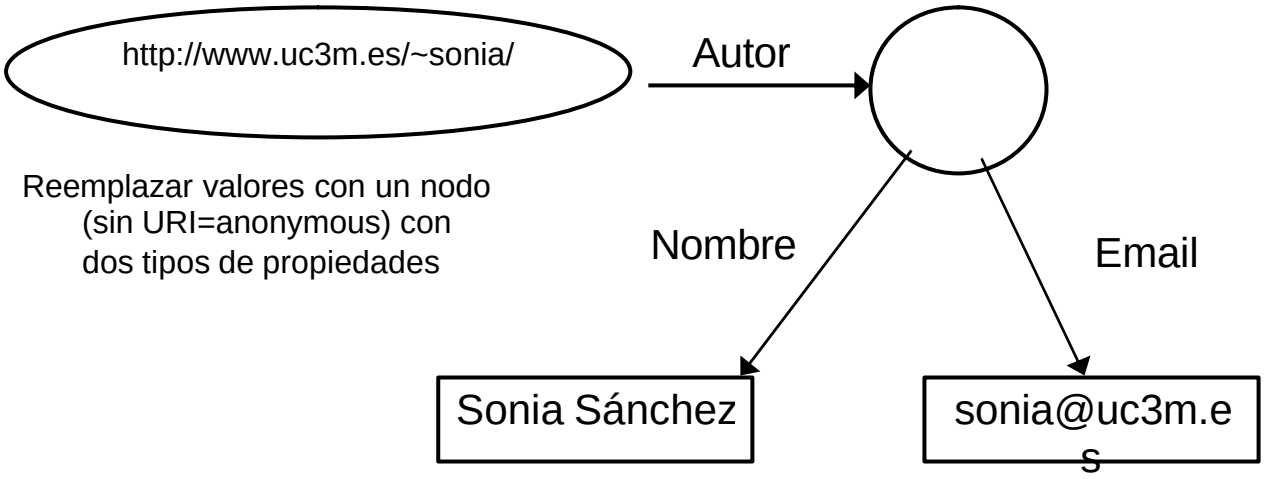


"Pepe es el autor del recurso identificado por http://www.uc3m.es/~pepe/"
Sujeto/Resource: la URI ; Verbo/Propiedad: autor
Predicado/Values: Pepe Pérez
<rdf:description rdf:about="www.uc3m.es" dc:creator="Pepe">

Ejemplo de RDF

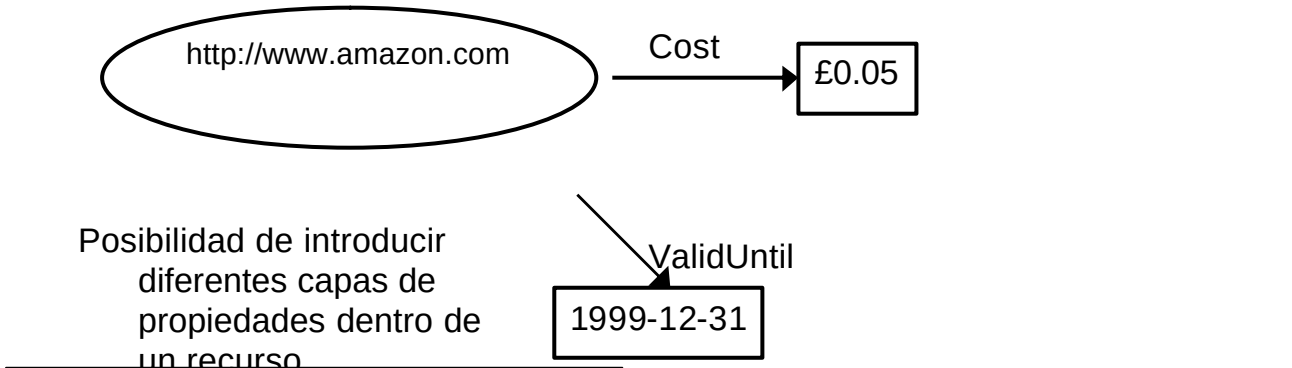
```
<?xml version="1.0" ?>
<RDF xmlns = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#" xmlns:DC = "http://purl.org/DC#">
<Description about = "http://www.amazon.com" >
        <DC:Title> Ontologia </DC:Title>
        <DC:Creator> Ruben Prieto-Diaz </DC:Creator>
        <DC:Date> 1999-12-31 </DC:Date>
        <DC:Subject> Metadata, RDF, Dublin Core </DC:Subject>
</Description>
</RDF>
```

RDF - estructuración




```
<DC:Creator parseType="Resource">
   <vCard:FN> Pepe </vCard:FN>
   <vCard:TITLE> profe </vCard:TITLE>
   <vCard:EMAIL> sonia@uc3m.es
</vCard:EMAIL>
</DC:Creator>
...
```

RDF – reification




```
...
<Description about =
http://www.amazon.com
          bagID = "ID001" >
  <DC:Title> Ontologias </DC:Title>
  <DC:Creator> Ruben
       Prieto</DC:Creator>
  <ECOMM:Price>£0.05</ECOMM:Price>
</Description>
```

```
<Description aboutEach = "#ID001" >
  <ADMIN:ValidFrom> 1998-01-01
</ADMIN:ValidFrom>
  <ADMIN:ValidTo> 1999-12-31
</ADMIN:ValidTo>
</Description>
```

RDF – mútiples propiedades

```
...
<DC:Creator>
  <Bag>
    <li> Maddie Azzurii
</li>
    <li> Corky Brown </li>
    <li> Jacky Crystal </li>
  </Bag>
</DC:Creator>…...
```

```
...
<DC:Creator>
  <Seq>
    <li> Maddie Azzurii </li>
    <li> Corky Brown </li>
    <li> Jacky Crystal </li>
  </Seq>
</DC:Creator>
...
```

```
...
<DC:Creator>
  <Seq>
    <li> Maddie Azzurii </li>
    <li> Corky Brown </li>
    <li> Jacky Crystal </li>
  </Seq>
</DC:Creator>
...
```

RDF – namespaces

- Utilizados en XML para representar atributos
- Identifican Tipos de Propiedades
- Deben especificarse previamente
- Precedidos de dos puntos
    - <DC:Title>Título del recurso</DC:Title>
- Tienen asociados un URI

RDF – namespaces

- Utilizados en XML para representar atributos
- Identifican Tipos de Propiedades
- Deben especificarse previamente
- Precedidos de dos puntos
    - <DC:Title>Título del recurso</DC:Title>
- Tienen asociados un URI

Ejemplo RDF

<?xml version="1.0"
<RDF xmlns:="http://w3.org/TR/1999/PR-rdf-syntax-19990222#"
xmlns:DC="http://purl.org/DC#">

<Description about="http://www.ugr.es">
<DC:Title> Web de la Universidad de Granada
</DC:Title>
<DC:Creator>Servicio de informática </DC:Creator>
<DC:Date> 1998-02-08 </DC:Date>
<DC:Description> Resumen del contenido del

```
        sitio</DC:Description>
  </Description>
</RDF>
```

RDF con varios vocabularios de metadatos

```
<?xml version="1.0"
<RDF xmlns:="http://w3.org/TR/1999/PR-rdf-syntax-19990222#"
        xmlns:DC="http://purl.org/DC#">
        xmlns:AGLS="http://na.gov.au/AGLS#"

  <Description about="http://www.ugr.es">
        <DC:Title> Web de la Universidad de Granada
        </DC:Title>
        <DC:Creator>Servicio de informática        </DC:Creator>
        <DC:Date> 1998-02-08 </DC:Date>
        <AGLS:Function> Information managemen – Internet </AGLS:FunctionDescription>
  </Description>
</RDF>
```

¿Qué es RSS?

- RSS = "Really Simple Syndication"
- RSS = "Rich Site Summary"
- RSS = "RDF Site Summary".

- Básicamente: RSS es un lenguaje XML para sindicar *(sindicate)* noticias en Internet.

Posibilidades

- Noticias (prensa, anuncios)
- Eventos
- Información de proyectos
- Bibliografías
- Información de contacto
- …

- La principal ventaja: La fuente informa cuando se producen cambios.

Un poco más técnico ...

- Los archivos RSS se actualizan de forma regular y contienen **metadatos** sobre una fuente de noticias determinada y su contenido.

- Consta fundamentalmente de:

    - **Channel**: que representa la fuente de las noticias.
    - **Title**: titulo del canal.

- **Link**: vínculo del canal.
- **Description**: descripción del canal.

- Además, consta de uno o varios elementos **item** que representan elementos de noticias individuales, cada uno de los cuales debe disponer de un campo **title**, **link** o **description**.

Ejemplo

<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title> BCR: The Third Indicator</title>
<link>http://www.bcr.org/publications/thirdind/</link>
<description>The Third Indicator, published monthly, is a technical memo focusing on OCLC products and services. It includes general OCLC news as well as detailed technical information on cataloging, reference and resource sharing. Announcements of new OCLC developments are also included.</description>
<lastBuildDate>Tue, 21 Sep 2004 21:37:39 GMT</lastBuildDate>
<generator>ListGarden Program 1.01</generator>
<docs>http://blogs.law.harvard.edu/tech/rss</docs>
<item>
<title>WorldCat Resource Sharing Training</title>
<link>http://www.bcr.org/publications/thirdind/2004/august/augsharetrain04.html</link>
<description>If you'd like to see what WorldCat Resource Sharing looks like and learn more about it, visit the OCLC Web site at www.oclc.org/ill/migration/ or view the WorldCat Resource Sharing tutorial at www5.oclc.org/downloads/tutorials/firstsearch/sv/rsbasics/intro/index.html/.</description>
<pubDate>Tue, 21 Sep 2004 19:29:47 GMT</pubDate>
<guid isPermaLink="false">thirdind-2004-08-21-19-29-47</guid>
</item>
</channel>
</rss>

¿Cómo funciona RSS?

- El autor crea un fichero RSS.
- Los usuarios se suscriben al fichero a través de un lector de noticias o agregador.
- Cuando el autor actualiza el fichero RSS, los nuevos elementos se notifican automáticamente a los usuarios quedando a su disposición para su lectura.

¿Qué es un canal?

- Un canal, bitácora o *blog* (abreviatura de *weblog*, pronunciado "*we blog*",) es un conjunto de noticias *online*, representadas en orden cronológico inverso e incluidas en un fichero RSS.
- También se utiliza para denominar el sistema que aloja y sirve un conjunto de canales.
- Típicamente, este tipo de sistemas incluyen enlaces entre ellos, proporcionando información adicional sobre los temas que incluyen.

¿Qué es un canal?

- Un canal, bitácora o *blog* (abreviatura de *weblog*, pronunciado "*we blog*",) es un conjunto de noticias *online*, representadas en orden cronológico inverso e incluidas en un fichero RSS.
- También se utiliza para denominar el sistema que aloja y sirve un conjunto de canales.
- Típicamente, este tipo de sistemas incluyen enlaces entre ellos, proporcionando información adicional sobre los temas que incluyen.

¿Qué es un agregador?

- Una aplicación o un servicio remoto que, periódicamente, lee un conjunto de fuentes o canales en formato XML.

- Cuando detecta nuevos elementos, muestra un resumen de los mismos en un listado ordenado cronológicamente, comenzando por el más moderno.

- La aplicación necesaria para leer ficheros RSS.

Tipos de agregadores

- Clientes/agentes independientes
  - **FeedReader**, Radio UserLand
- Complementos PIM *(Personal Information Manager)*
  - **Pluck,** NewsGator, intraVnews
- Complementos de Navegador
  - **Firefox 1.0**, Sage
- Sitios Web
  - **Bloglines**, NewsIsFree

- Listado de agregadores
  http://www.lights.com/weblogs/rss.html

Mas información

- Introducción a RSS http://www.maestrosdelweb.com/editorial/sindicacion/
- RSS esquema http://www.clikear.com/xsd/rss2.xsd
- RDF Site Summary http://web.resource.org/rss/1.0/spec

- cANALES

- Librarian.net www.librarian.net
- Librarian's Rant lblog.jalcorn.net
- LISNews www.lisnews.com
- The Shifted Librarian www.theshiftedlibrarian.com
- Travelin' Librarian travelinlibrarian.blogspot.com
- Unshelved www.overduemedia.com
- Free Range Librarian freerangelibrarian.com
- Crime in the Library crimeinthelibrary.blogspot.com
- Tame the Web www.tametheweb.com/ttwblog
- LibraryTechtonics www.librarytechtonics.inf

TEMA 4: SQL CON EJEMPLOS

SQL

- Como DDL nos permite Crear y borrar tablas y relaciones (mediante CREATE, DROP y ALTER).
- Como DML están SELECT (selección registros), UPDATE (actualizar registros), DELETE (borrar registros) e INSERT (añadir registros). Sirve para que consultemos y modifiquemos los datos.
- Como Lenguaje de Control. GRANT (para dar privilegios), REVOKE (quitar privilegios), EXPLAIN y LOCK. Que sirven para controlar el acceso a las tablas.
- Guía de referencia en https://aurora.vcu.edu/db2help/db2s0/frm3toc.htm

Utilidades

- Estándar ISO y ANSI
- Es el lenguaje más universal existente para trabajar con BD
- Se puede insertar dentro del código de la mayoría de lenguajes de programación para así acceder a datos de BD (Visual C, .Net, ...) [Forma inmersa en un lenguaje anfitrión]
- Se puede emplear dentro de cualquier base de datos relacional actual (Oracle, Access, SQL Server...)
  [existen otros lenguajes como QBE, QUEL...]
- Es sencillo
- Muchas consultas no se pueden realizar en la ventana de diseño de consultas

CONSULTAS DE DEFINICIÓN DE DATOS. CREACIÓN DE TABLAS

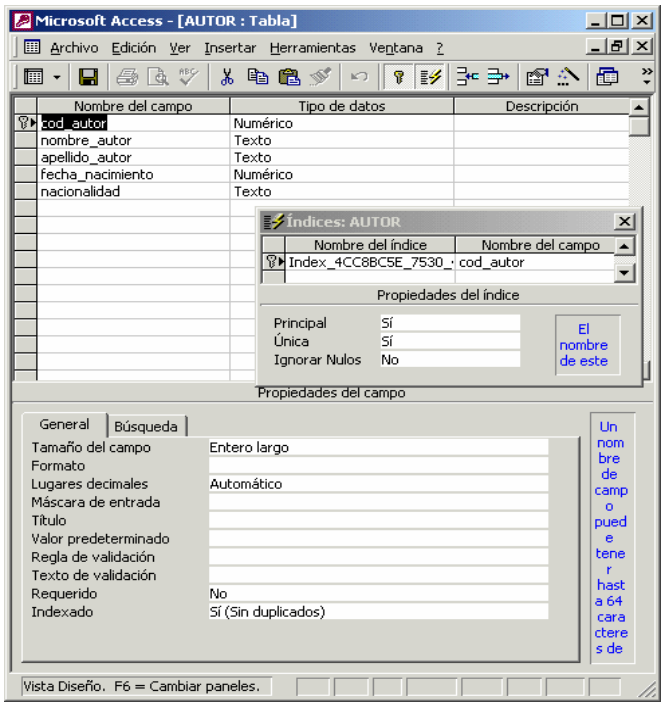
El comando CREATE sirve para crear una tabla nueva.

CREATE TABLE nombre-tabla-nueva
(nombre-campo1 tipoDato otrasPropiedades, nombre-campo2 tipoDato otrasPropiedades, …)

TipoDato: Integer, String, char, bit, date, real, etc
Otras propiedades: tamaño del campo (número caracteres entre paréntesis si es string), PRIMARY KEY, Not null, …

CONSULTAS DE DEFINICIÓN DE DATOS. CREACIÓN DE TABLAS

CREATE TABLE AUTOR (cod_autor integer PRIMARY KEY, nombre_autor text (70) not null, apellido_autor text (70), fecha_nacimiento integer, nacionalidad text (50))



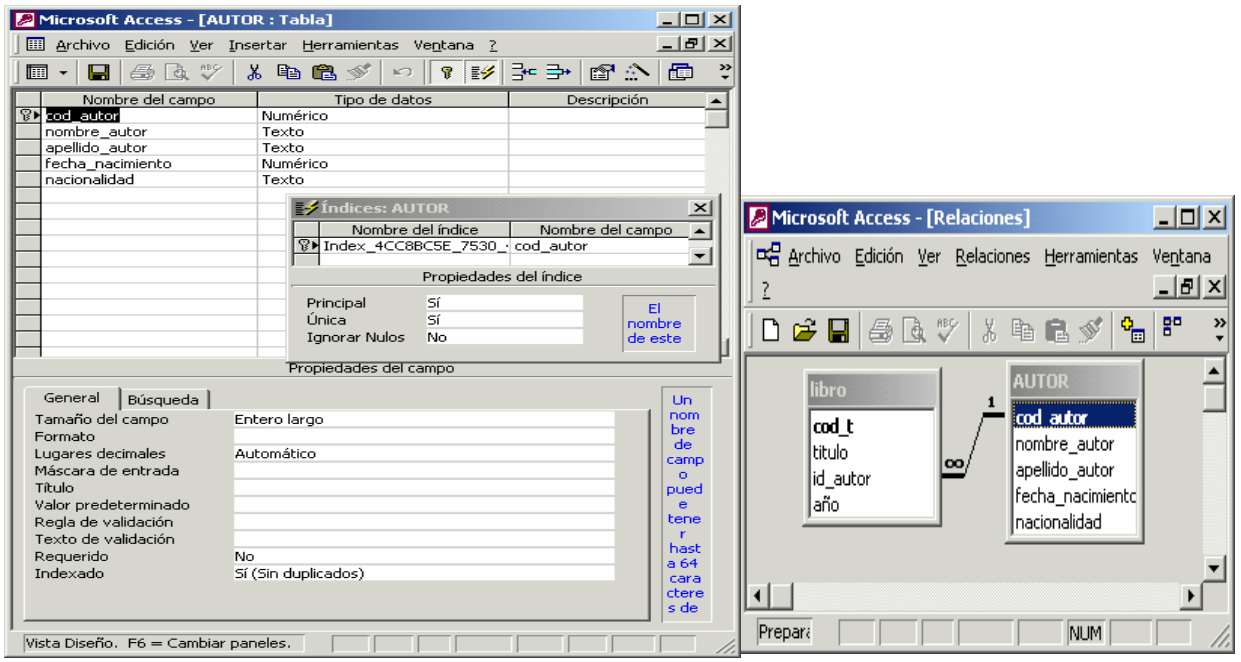
CREAR UNA RELACIÓN ENTRE TABLAS CON CREATE

Las relaciones entre tablas suelen ser entre una primary key (clave principal) y una foreign key (clave ajena)

CREATE TABLE nombre-tabla
(nombre-campo1 tipoDato propiedad, nombre-campo2 tipoDato propiedad,…,
CONSTRAINT nombre_clave
FOREIGN KEY (campo_clave_ajena)
REFERENCES tabla-a-relacionar (campo-de-la-tabla-a-relacionar))

CONSULTAS DE DEFINICIÓN DE DATOS. CREACIÓN DE TABLAS

CREATE TABLE LIBRO (cod_t integer primary key, titulo text (70) not null, id_autor integer, año integer, CONSTRAINT f FOREIGN KEY (id_autor)
REFERENCES autor(cod_autor))



Crear una Tabla Nueva

- Los tipos de datos pueden ser: text (o string), date, si/no (bit), número (integer, float, real,etc), moneda (currency)...

A partir de otra tabla tb se puede crear una nueva tabla:

- SELECT campo1_origen[, campo2[, ...]] INTO nuevatabla [IN basededatosexterna] FROM tabla_origen

DDL. MODIFICAR LA ESTRUCTURA DE UNA TABLA

- El COMANDO ALTER TABLE sirve para añadir, modificar, eliminar campos y claves de una tabla

ALTER TABLE tabla-a-modificar
ADD/ALTER/DROP COLUMN campo

ADD añade columna, DROP la elimina y ALTER COLUMN modifica su tipo de datos o tamaño

DDL. MODIFICAR LA ESTRUCTURA DE UNA TABLA

- **Añadir columna idioma:**

ALTER TABLE libro
ADD COLUMN idioma text (15)

- **Añadir columna lugar:**

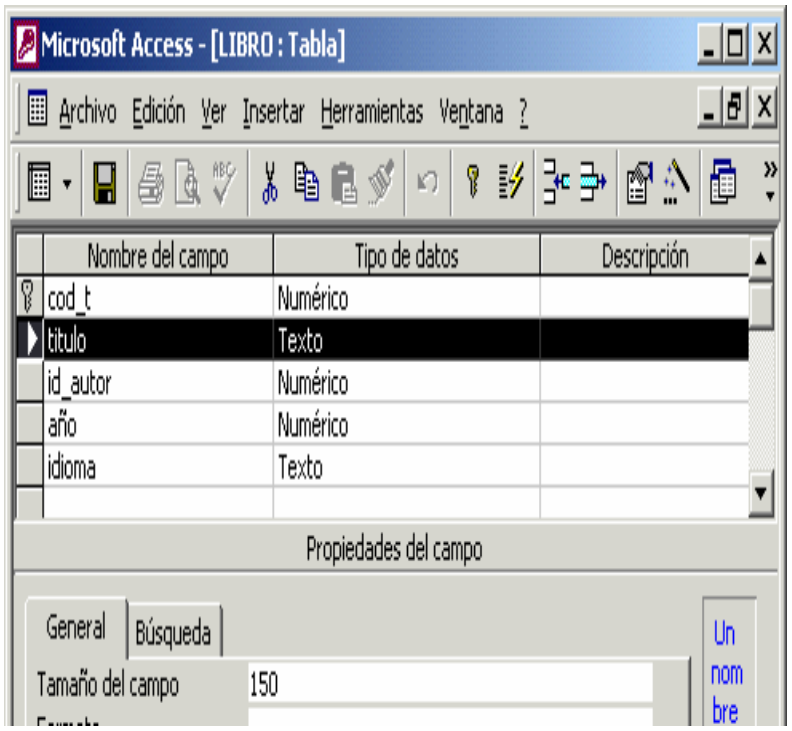
ALTER TABLE libro
ADD COLUMN lugar text (15)

- **Modificar el tamaño de la columna titulo:**

ALTER TABLE libro
ALTER COLUMN titulo text (150)

- **Eliminar la columna lugar:**

ALTER TABLE libro
DROP COLUMN lugar



Modificar y Eliminar una tabla
Crear índices

MODIFICAR UNA TABLA

- ALTER TABLE tabla {ADD {COLUMN campo tipo [(tamaño)] [NOT NULL] [CONSTRAINT índice] |CONSTRAINT índicemúltiplescampos} |DROP {COLUMN campoI CONSTRAINT nombreíndice} }

ELIMINAR UNA TABLA O INDICE

- DROP {TABLE tabla | INDEX índice ON tabla}

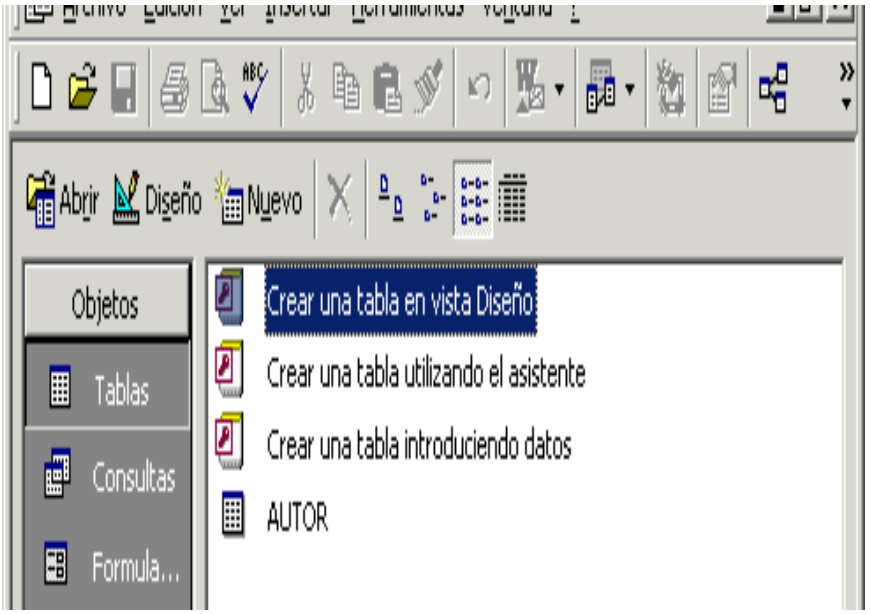
CREAR INDICES

- CREATE [UNIQUE] INDEX índice ON tabla(campo [ASC|DESC][, campo [ASC|DESC], ...]) [WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }]

DDL. ELIMINAR UNA TABLA

- **DROP TABLE**
  tabla-a-eliminar

- **DROP TABLE**
  Libro



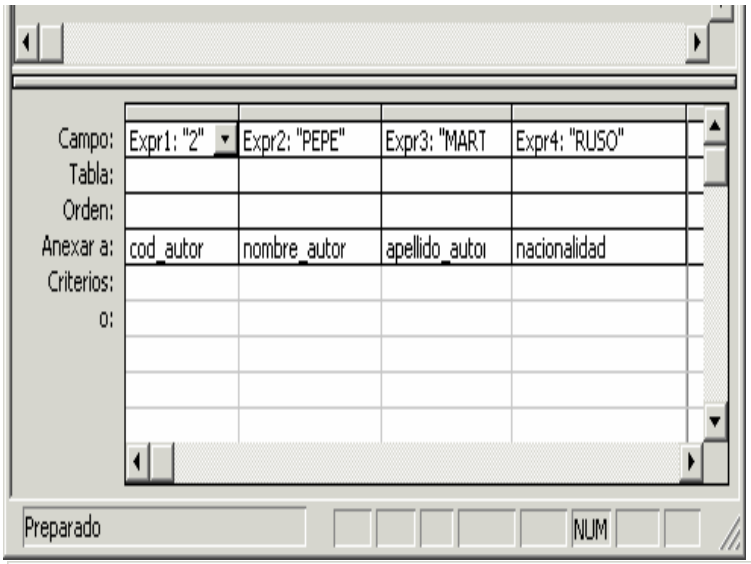
INSERT

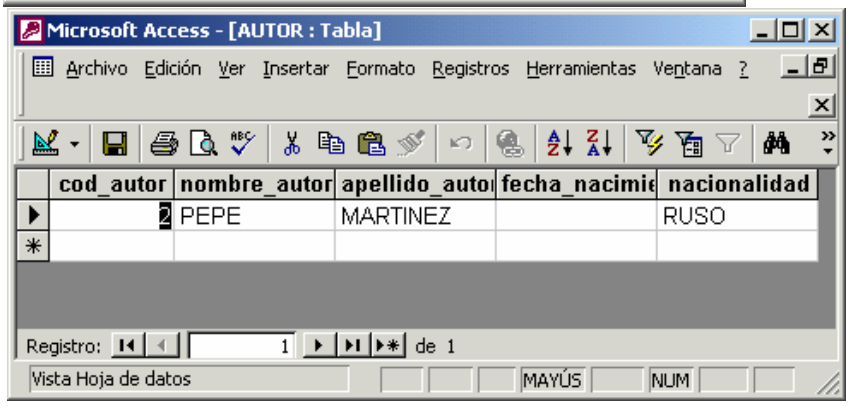
- Sirve para anexar datos, esto es añadir una nueva fila con datos a determinada tabla

INSERT INTO Tabla-a-anexar
(campos-de-la-tabla-a-rellenar)
VALUES (valores con los que completar los campos);

INSERT INTO SALARIO
(CA_PERSONAL, SUELDO, MES, AÑO)
VALUES (30, 128000, "Diciembre", "2004“);

INSERT INTO AUTOR
(cod_autor,nombre_autor,apellido_autor,nacionalidad)
VALUES ("2","PEPE","MARTINEZ","RUSO")





UPDATE (ACTUALIZACIÓNES)

- Cambia el valor de una o varias celdas por un nuevo valor

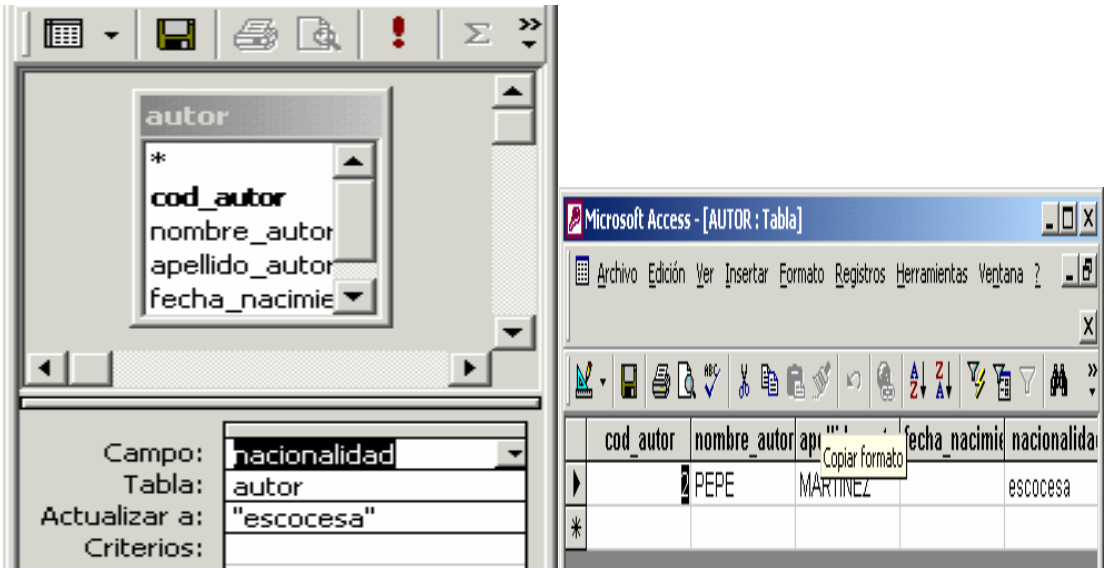
UPDATE TABLA-A-ACTUALIZAR
SET CAMPO-A-ACTUALIZAR=“VALOR-NUEVO”
WHERE CAMPO-A-ACTUALIZAR=“VALOR ANTIGUO”

- Ejemplo para actualizar la tabla Salario y poner en el campo CATEGORIA el valor jefe siempre que aparezca la palabra consejero

UPDATE SALARIO
SET CATEGORIA = "jefe"
WHERE CATEGORIA="consejero"

UPDATE

UPDATE autor
SET autor.nacionalidad = ‘escocesa’
WHERE autor.nacionalidad)=’ruso’



RESUMEN ACTUALIZACIÓN

- UPDATE tabla SET campo=loquesea WHERE criterio;

Si hay varias tablas:

- UPDATE tabla1 INNER JOIN tabla2 ON tabla1.campo1=tabla2.campo2 SET campo=loquesea WHERE criterio

DELETE (BORRAR)

Sirve para eliminar los registros que cumplan alguna condición
DELETE CAMPO-CUYO-VALOR-SE-QUIERE-BORRAR
FROM TABLA-QUE-CONTIENE-EL-CAMPO
WHERE CAMPO-CUYO-VALOR-SE-ELIMINA-SI-TIENE-CIERTO-VALOR=VALOR

Por ejemplo para eliminar en la tabla salarios los registros relacionados con Ana García será:
DELETE SALARIO.*
FROM PERSONAL
INNER JOIN SALARIO
ON PERSONAL.COD = SALARIO.CA_PERSONAL
WHERE PERSONAL.NOMBRE="García, Ana“

O en la tabla autor:
DELETE * FROM autor WHERE nacionalidad='escocesa'

BD DE EJEMPLO

COD	NOMBRE	DNI	FECHA	SALARIO	SEX
10	Hernandez, Cris	34636321	651001		F
20	Tapia, Miguel	55789642	731010		M
30	García, Ana	20389571	750405	38250	F

Tabla PERSONAL

Tabla SALARIO

CA_PER	SUELDO	MES	AÑO	CATEGORIA
10	150253	Agosto	2004	Administrativo
10	120000	Septiembre	2004	Administrativo
10	120000	Ocutbre	2004	Administrativo
20	450000	Septiembre	2004	Consejero
20	450000	Octubre	2004	Consejero
30	120000	Julio	2004	Administrativo
30	150253	Agosto	2004	Administrativo
30	650000	Septiembre	2004	Consejero

SELECT

SELECT campo1,campo2

Obligatorio. Pon los campos que quieres ver como resultado de la consulta separados por comas, si todos →*

FROM tabla

Pon las tablas donde están los campos, si varias pon comas

Campos opcionales

WHERE condición

Condición(-es) que deben cumplir los registros que visualices. Si coexisten AND y OR,... usa paréntesis. Si es campo no numérico pon contenido entre comillas. Si pones comodines pon LIKE

GROUP BY campo

Agrupar por un campo o campos

ORDER BY campo

Ordenar por un campo o campos ASC ascendente o DESC inverso

HAVING condición
Condición tras agrupar

FUNCIONES DE AGREGADO Y VALORES DE WHERE

FUNCIONES DE AGREGADO
AVG (media), COUNT (contar), SUM (sumar), MAX (el máximo), MIN (el mínimo)

CAMPO WHERE

- LIKE “texto*”
- LIKE “texto?”
- =numero (tb >=, < =, <>(distinto))
- =”texto”
- BETWEEN A AND B
- IS NOT NULL/ IS NULL
- Normalmente si queremos negar una situación se usa NOT tras el nombre del campo y luego la condición
- Para combinar varias condiciones en el WHERE se usan paréntesis y operadores booleanos (AND, OR)

EJEMPLOS SELECT

- Selecciona todos los campos y todos los registros de la tabla personal
  SELECT * FROM PERSONAL

- Selecciona los campos cod, nombre y fecha y todos los registros
  SELECT cod,nombre,fecha FROM PERSONAL

- Selecciona nombre y DNI de las empleadas de la empresa
  SELECT nombre,DNI FROM PERSONAL WHERE SEXO=“F”

- Selecciona los empleados con el campo salario de la tabla PERSONAL vacío
  SELECT nombre,DNI FROM PERSONAL WHERE SALARIO IS NULL

- Selecciona todos los empleados cuyo apellido comience por T
  SELECT * FROM PERSONAL WHERE NOMBRE LIKE “T*”

EJEMPLOS SELECT

- Selecciona los empleados cuyo nombre contenga una “e” o que sean mujeres
  SELECT * FROM PERSONAL WHERE NOMBRE LIKE “M*” OR
  SEXO=“F”

- Selecciona de la tabla salario los sueldos mayor que 200000 y distintos de 450000 que no pertenezcan al mes de Agosto
  SELECT sueldo FROM SALARIO WHERE SUELDO > 200000 AND
  SUELDO<>450000

- Selecciona de la tabla salario los sueldos entre 100000 y 150000 que no pertenezcan al mes de Agosto
  SELECT sueldo FROM SALARIO WHERE SUELDO BETWEEN 100000 AND 150000 AND MES NOT LIKE “AGOSTO”

- Selecciona los empleados con el campo salario de la tabla PERSONAL no este vacío
  SELECT nombre,DNI FROM PERSONAL WHERE SALARIO IS NOT NULL

EJEMPLOS SELECT

- Ordena a los empleados por nombre ascendente
  SELECT * FROM PERSONAL ORDER BY NOMBRE ASC

- Selecciona los empleados cuyo nombre contenga la palabra garcía y que sean mujeres o cuyo DNI sea 55789642 ordena por numero de DNI descendente
  SELECT * FROM PERSONAL WHERE ((NOMBRE LIKE ‘*GARCÍA*' AND SEX='F') OR DNI=55789642) ORDER BY DNI DESC

- Mostrar en una sola fila y sin duplicados los meses distintos que aparecen en la tabla salarios
  SELECT DISTINCT mes FROM SALARIO

SELECT FUNCIONES Y AGRUPAMIENTOS

SELECT campo1
Función de agrupamiento, para calcular la media (AVG), suma (SUM), contar (COUNT), valor mínimo (MIN), máximo (MAX)...

AVG(campo2)
FROM tabla

WHERE condición
Opcional. Condición antes de agrupar

GROUP BY campo
Agrupar por un campo o campos, cuando se ha puesto una función de agrupamiento en el SELECT todos los campos sin función deben estar agrupados. Si varios se separan por comas

ORDER BY campo
HAVING condición
Opcional. Condición tras agrupar

EJEMPLOS

- Suma de lo pagado en el mes de Agosto
  SELECT Sum(SUELDO) FROM SALARIO WHERE MES="agosto"

- Media de lo pagado a los empleados cada mes. El campo calculado deberá llamarse media

```
SELECT Avg(SALARIO.SUELDO) AS Media, SALARIO.MES
FROM SALARIO
GROUP BY SALARIO ME
```

EJEMPLOS

- Contar el número de salarios que se pagaron en octubre del 2004

```
AÑO,MES, Count(MES) AS Cuenta
FROM SALARIO
WHERE AÑO=2004 AND MES=“octubre”
GROUP BY AÑO, MES
```

- Que sueldo cobraron los empleados que ganaron más de 800000 en el 2004

```
SELECT Sum(SUELDO) AS Suma, AÑO, CA_PERSONAL
FROM SALARIO
WHERE AÑO=2004
GROUP BY AÑO, CA_PERSONAL
HAVING SUM(SUELDO)>=800000
```

Notas: Evitar campos ambiguos

- Si en vez del nombre del campo pones el nombre de la tabla un punto y nombre del campo quedará menos ambiguo
  autor.titulo en vez de titulo

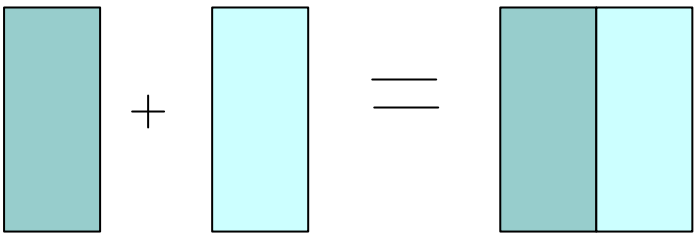
CONSULTA DE UNIÓN

- Requisitos: deben de tener igual estructura las dos tablas
- La consulta unión sirven para ver en un único campo los datos de igual tipo de varias tablas. En Access se realiza en diseño de consultas, menú consulta, opción específica de SQL y Unión. O directamente escribiéndola en la pantalla de SQL. No se puede hacer en modo gráfico.
- SELECT [CAMPO1], [CAMPO2] FROM [TABLA1] UNION SELECT [CAMPO1], [CAMPO2] FROM [TABLA2];
- La TABLA1 y la TABLA2 deben de tener el mismo número de campos.
- El resultado es que devuelve en una misma columna el resultado de las dos tablas.

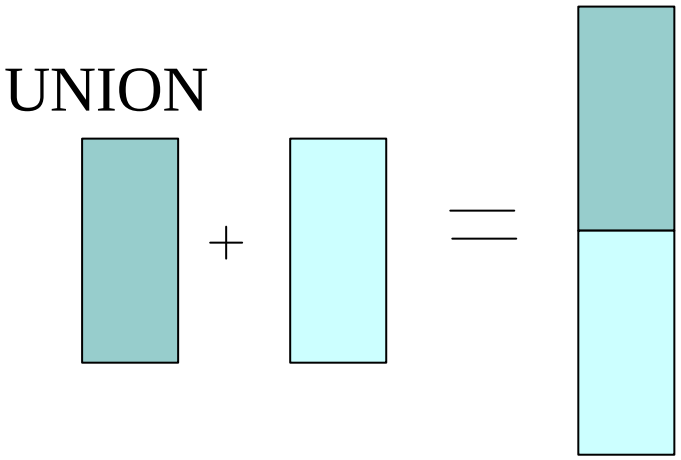
- Los registros duplicados se eliminan. Si no se quiere que se eliminen se escribe UNION ALL

JOIN vs UNION

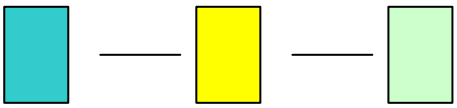
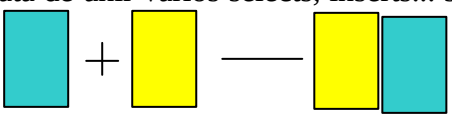
JOIN



UNION



CONSULTAS A PARTIR DE VARIAS TABLAS

- Cuando se unen varias tablas es mejor poner el nombre completo (tabla.campo) para evitar ambigüedades
- CON SUBCONSULTAS 
  - Menos trabajo para el ordenador
  - Se trata de unir varios selects, inserts... seguidos
- CON JOIN 
  - Necesario si se quiere mostrar campos de varias tablas simultáneamente
  - Se trata de unir varias tablas en una a partir de campos del mismo tipo (usualmente, aunque no siempre, con clave principal-clave ajena)

JOIN

- Mostrar en una consulta el nombre de los empleados junto con la remuneración en el 2004
- Existen dos formas equivalentes
  - CON INNER JOIN...ON

```
SELECT PERSONAL.NOMBRE, Sum(SALARIO.SUELDO) AS Remuneracion FROM
PERSONAL INNER JOIN SALARIO ON PERSONAL.COD =
SALARIO.CA_PERSONAL WHERE SALARIO.AÑO=2004 GROUP BY
PERSONAL.NOMBRE;
```

  - PONIENDO LA INFORMACIÓN DE UNIÓN EN EL WHERE

```
SELECT PERSONAL.NOMBRE, Sum(SALARIO.SUELDO) AS Remuneracion
FROM PERSONAL,SALARIO WHERE PERSONAL.COD =
SALARIO.CA_PERSONAL AND SALARIO.AÑO=2004
GROUP BY PERSONAL.NOMBRE;
```

SUBCONSULTAS

- Mostrar en una consulta el nombre de los empleados que cobraron un sueldo en algún mes del 2004 superior a 200000 y que son mujeres
  - **Primero** tendré que ver en la tabla SALARIO que sueldos fueron en el 2004 superiores a 200000 y retener el valor del CA_PERSONAL. Si lo hago "a mano" puedo comprobar que los empleados con el CA_PERSONAL igual a 20 y a 30 cobraron
  - **Segundo** tendré que ver en la tabla PERSONAL que nombres tienen los empleados cuyo campo COD tiene los números 20 y 30 y cuyo campo sexo es igual a "f"

La forma de hacer la subconsulta es invirtiendo el orden anterior, primero pondremos el paso segundo y después el primero, de la siguiente forma:

SUBCONSULTA

- Mostrar en una consulta el nombre de los empleados que cobraron un sueldo en el 2004 superior a 200000 y que son mujeres

2

```
SELECT NOMBRE, SEX
FROM PERSONAL
WHERE COD IN ( * Esto indica que los registros resultado de la sentencia entre
                                  paréntesis se trasfieren al WHERE superior
```

1

```
SELECT CA_PERSONAL
FROM SALARIO
WHERE SUELDO>200000)
```

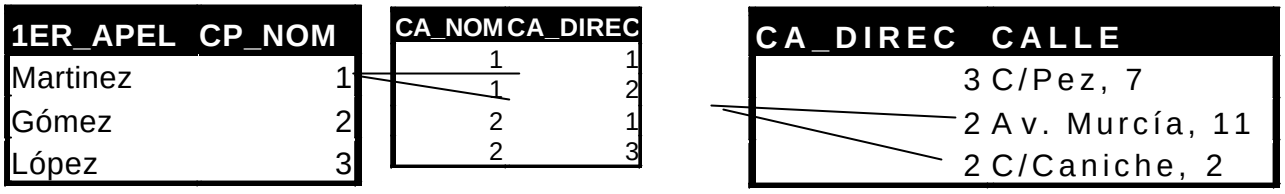
Subselect

```
AND SEX="F"
```

- Aquí no es necesario el nombre completo de los campos pues no hay ambigüedad
- La forma de pasar valores de una sentencia a otra es con:
  ...WHERE campo IN (...

Ejemplo de subconsulta

NOMBRE DIRECCION-NOMBRE DIRECCION



Q: SELECCIONAR LOS APELLIDOS DE LAS PERSONAS Q VIVAN EN LA CALLE CANICHE
Sin subconsultas habría que hacer tres selects:

1º CONSULTAR EN LA TABLA DIRECCIÓN LA CA_DIRECCIÓN CUANDO CALLE ES AV.MURCIA
SELECT CA_DIREC FROM DIRECCION WHERE CALLE LIKE'*caniche*'
RESULTADO=2

Ejemplo de subconsulta

2º CONSULTAR EN LA TABLA DIRECCIÓN-NOMBRE LA CA_NOMBRE CUANDO CA_DIRECCION ES 2
SELECT CA_NOM FROM DIRECCION-NOMBRE WHERE CA_DIREC=2
RESULTADO =1

3º CONSULTAR EN LA TABLA NOMBRE EL APELLIDO CUANDO CP_NOMBRE ES 1
SELECT 1ER_APEL FROM NOMBRE WHERE CP_NOM=1
RESULTADO = MARTINEZ

EN UNA SOLA CONSULTA CON SUBCONSULTAS
SELECT 1ER_APEL FROM NOMBRE WHERE CP_NOM IN (SELECT CA_NOM FROM DIRECCION-NOMBRE WHERE CA_DIREC IN (SELECT CA_DIREC FROM DIRECCION WHERE CALLE LIKE'*caniche*'))

SUBCONSULTA vs JOIN

- Muchas consultas se pueden realizar indistintamente con JOIN y SUBCONSULTAS
- JOIN es como hace la unión de tablas ACCESS por defecto

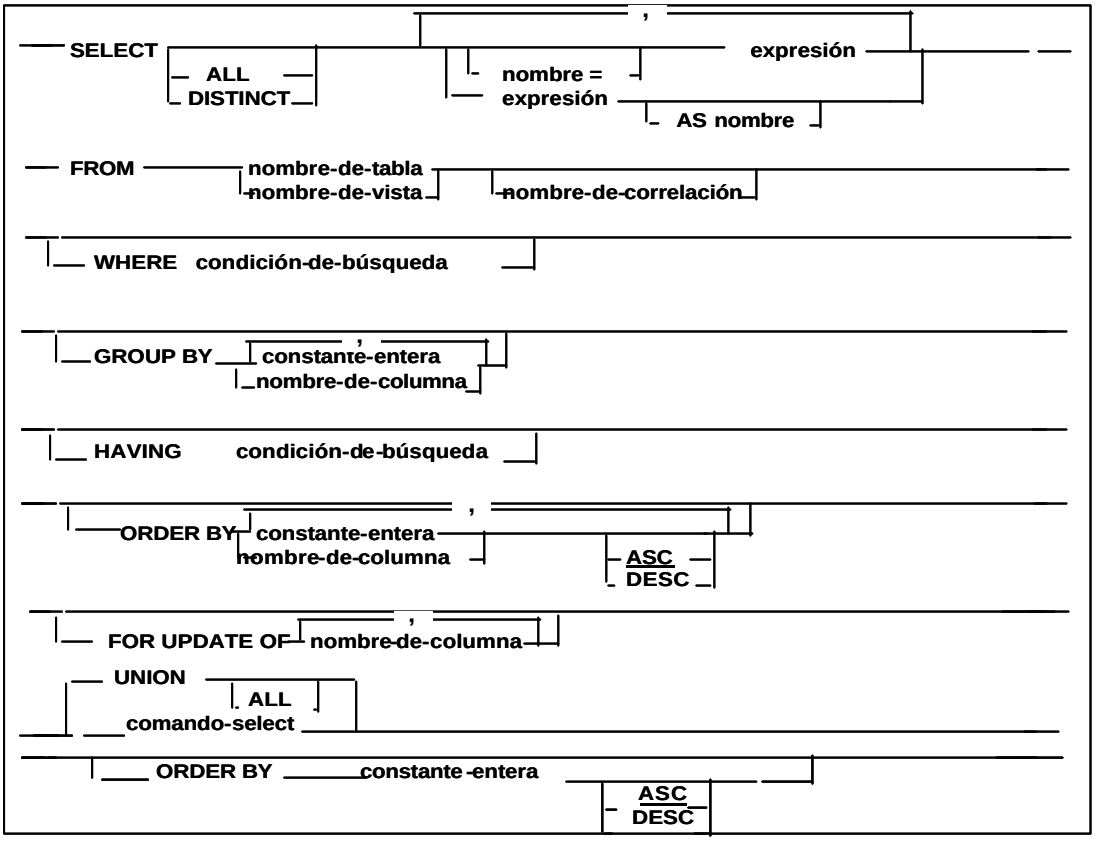
- Existe un caso en el que no se puede emplear subconsultas en vez de JOIN y es cuando nos piden que como resultado mostremos el contenido de varios campos procedentes de distintas tablas

NOTAS: CASILLAS EN BLANCO

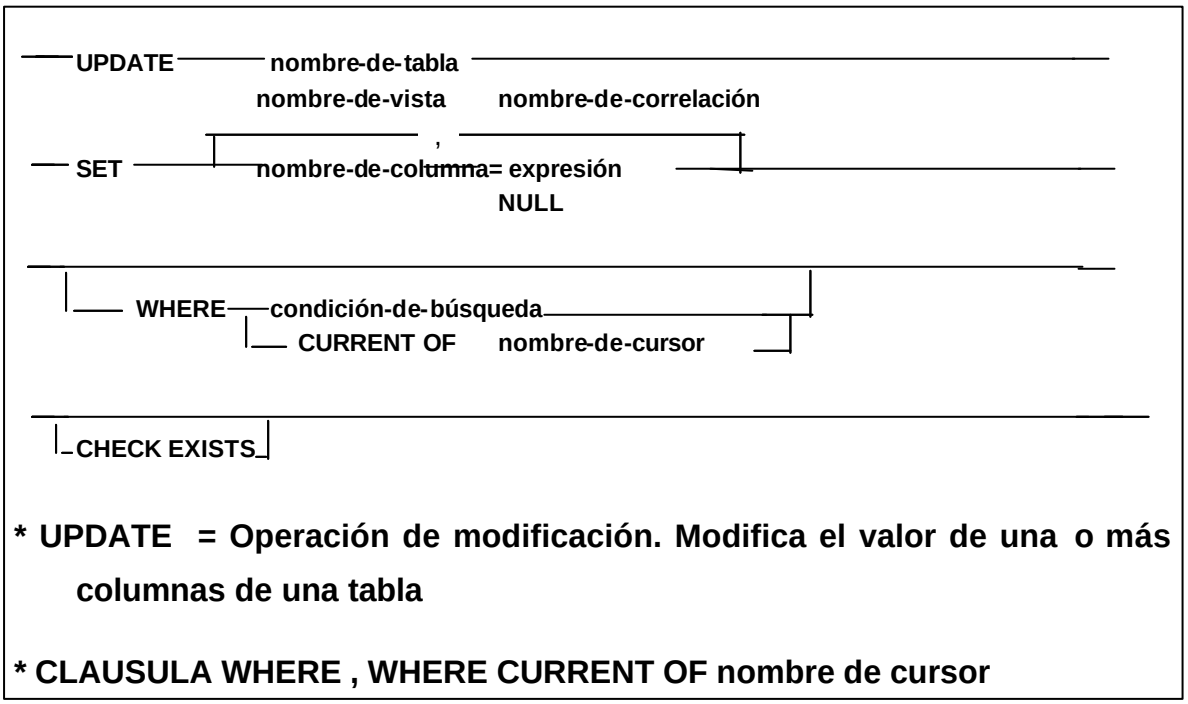
Puede haber dos razones para que una casilla este en blanco.

	Causa	Por defecto Access	Forma de rellenar la celda	Búsqueda
NULO	Se desconoce el valor	Permitido	No se inserta nada	Where <campo> is null
CADENA LONGITUD CERO	No existe el valor	No permitido	Se insertan dos comillas seguidas “”	Where <campo> = “”

SELECT



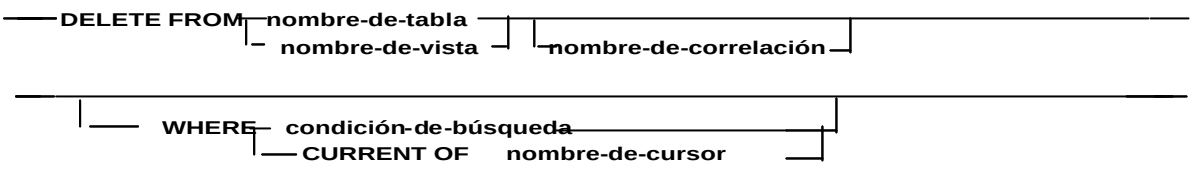
UPDATE



*** UPDATE = Operación de modificación. Modifica el valor de una o más columnas de una tabla**

*** CLAUSULA WHERE , WHERE CURRENT OF nombre de cursor**

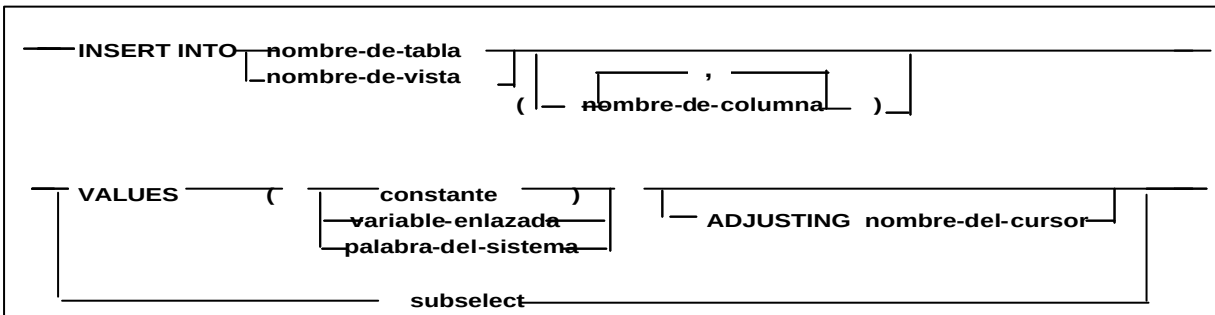
DELETE



*** DELETE = Operación de modificación. Elimina una o más filas d e una tabla**

*** CLAUSULA WHERE , WHERE CURRENT OF nombre de cursor**

INSERT



*** INSERT = Operación de Inserción. Añade una o más filas a una tabla**